Image Filtering

Dr. Tushar Sandhan

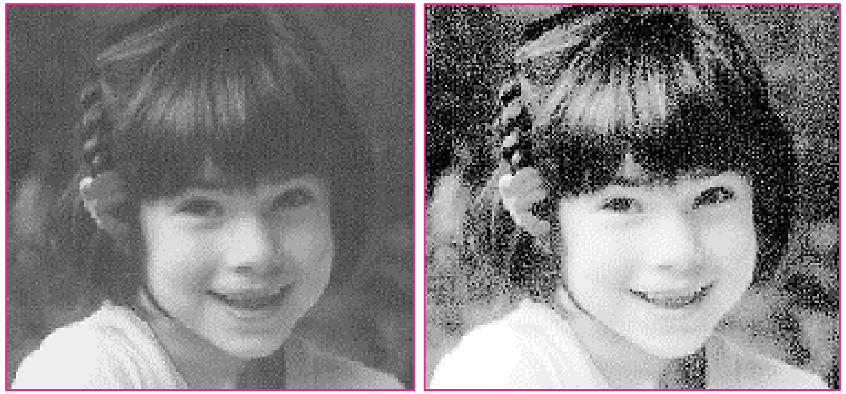
sandhan@iitk.ac.in

Introduction

Input

Introduction

Input



Introduction

Input

Histeq

Introduction

Input

Histeq

Introduction

Input

Histeq

Noise

Introduction

Input

Histeq

Noise

Introduction

Input

Histeq

Noise

Filter-1

Introduction

Input Histeq Noise Filter-1





Introduction

Input

Histeq

Noise

Filter-1

Filter-2

Introduction

Input Histeq Noise Filter-1 Filter-2

Introduction

Input Histeq Noise Filter-1 Filter-2 Filter-3

Introduction

Input

Histeq

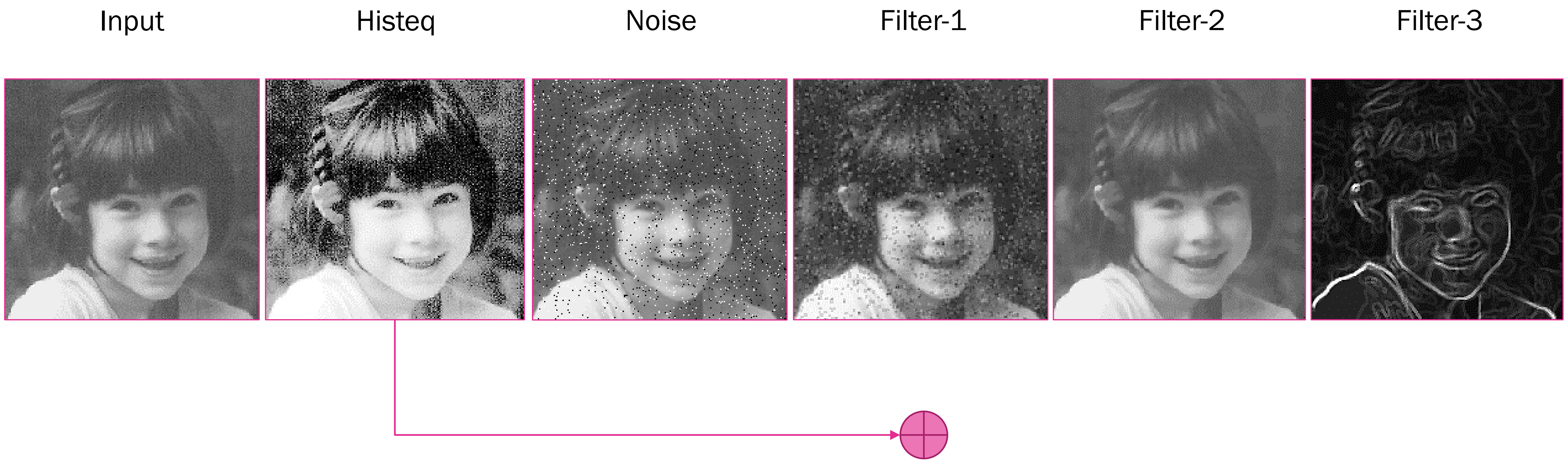
Noise

Filter-1

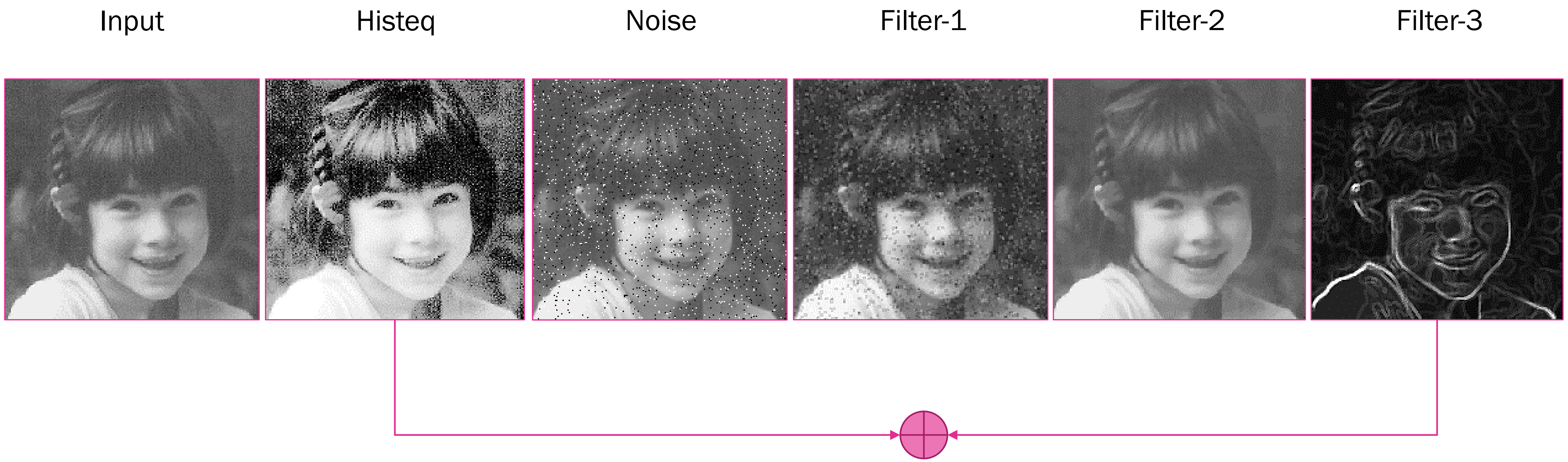
Filter-2

Filter-3

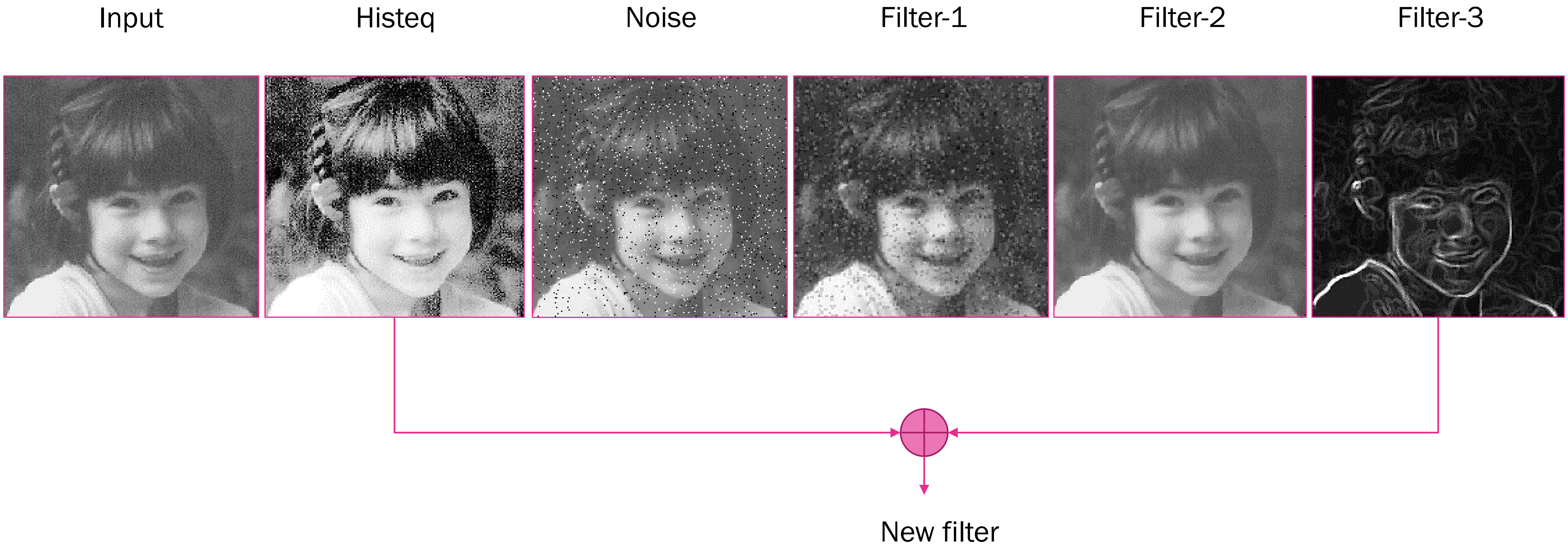
Introduction



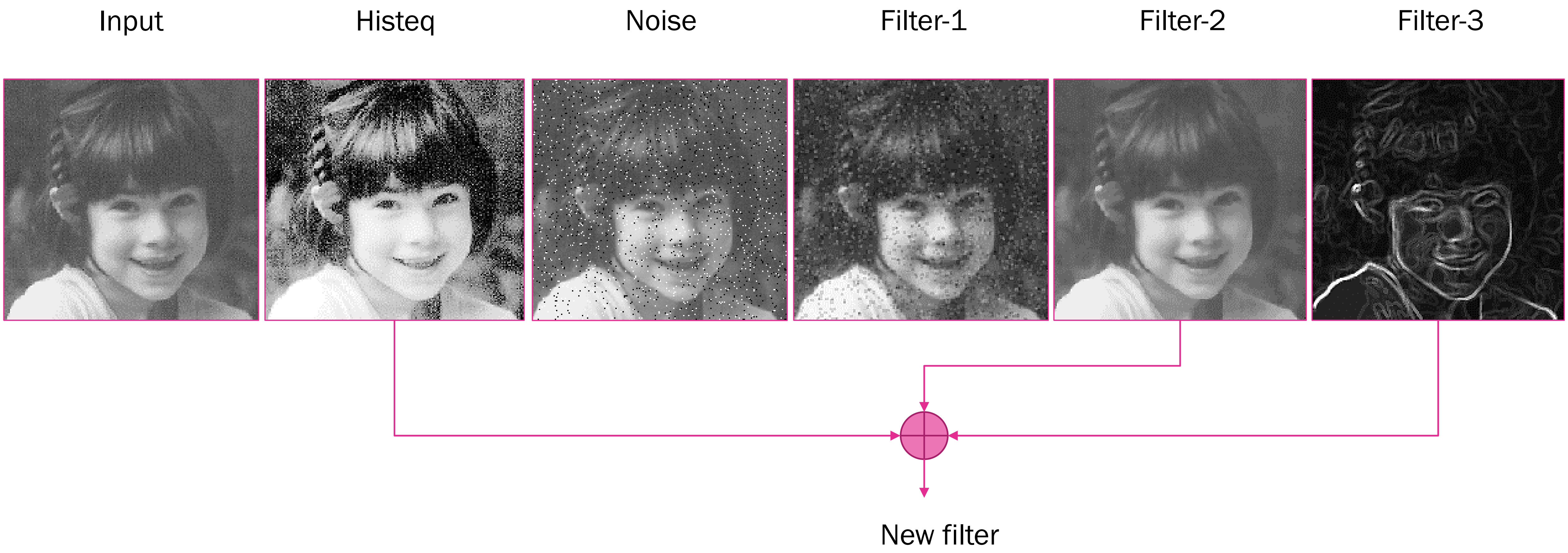
Introduction



Introduction



Introduction



Linearity

- Operations
  - linear
    - additivity

$$\mathcal{T}[a \cdot f_1(x,y) + b \cdot f_2(x,y)] = \mathcal{T}[a \cdot f_1\,(x,y)] + \mathcal{T}[b \cdot f_2\,(x,y)]$$

    - homogeneity

$$\mathcal{T}[a \cdot f_1(x,y)] = a \cdot \mathcal{T}[f_1(x,y)]$$

  - non-linear
    - not satisfying above

Linearity

- Operations
  - linear
    - additivity

$$\mathcal{T}[a \cdot f_1(x,y) + b \cdot f_2(x,y)] = \mathcal{T}[a \cdot f_1\,(x,y)] + \mathcal{T}[b \cdot f_2\,(x,y)]$$

    - homogeneity

$$\mathcal{T}[a \cdot f_1(x,y)] = a \cdot \mathcal{T}[f_1(x,y)]$$

  - non-linear
    - not satisfying above

- Examples
  - linear
    - negatives
  - non-linear
    - gammas

Correlation & Convolution

- Correlation
  - measures similarity between the two signals
  - windowed signal (kernel) is not reversed
  - sliding vectors dot product
  - orthogonal signals are uncorrelated

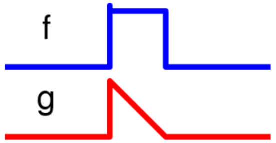
Correlation & Convolution

- Correlation
  - measures similarity between the two signals
  - windowed signal (kernel) is not reversed
  - sliding vectors dot product
  - orthogonal signals are uncorrelated

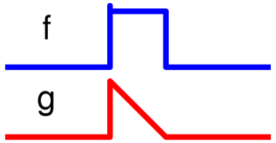
- Convolution
  - measure the effect of one signal on the another
  - windowed signal (kernel) is reversed
    - for symmetric kernels convolution = correlation

Correlation & Convolution

- Correlation



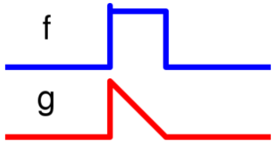
- Convolution



credit: wikipedia

Correlation & Convolution

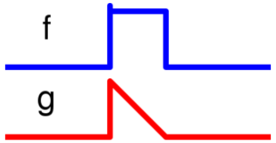
- Correlation



$$R(x) = f(x) \star g(x)$$

$$R(x) = \int_{-\infty}^{\infty} f(z) g(x+z) dz$$

- Convolution

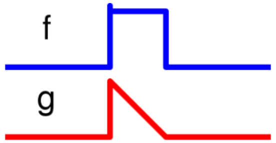


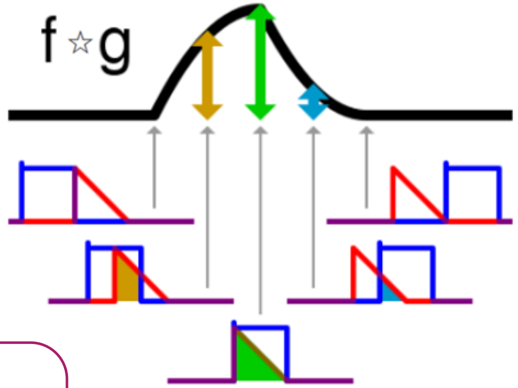
credit: wikipedia

‘ ⛤, ⊛, ✲, ✺, ✶, ✵, ✯, ✭, ✬, ☆, ✡, ✰, ✱, ✽, ✪ ’

Correlation & Convolution

- Correlation

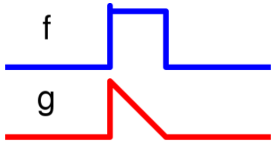




$$R(x) = f(x) \star g(x)$$

$$R(x) = \int_{-\infty}^{\infty} f(z) g(x+z) dz$$

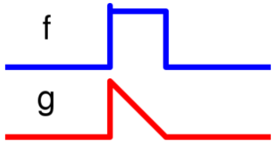
- Convolution

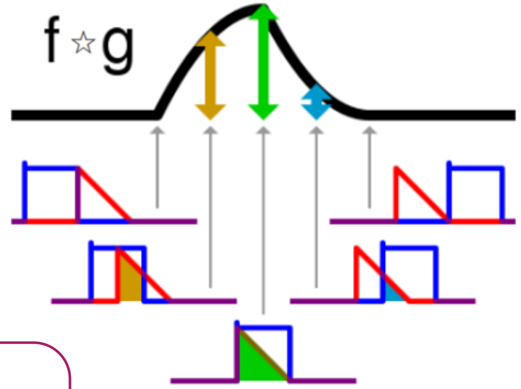


credit: wikipedia

Correlation & Convolution

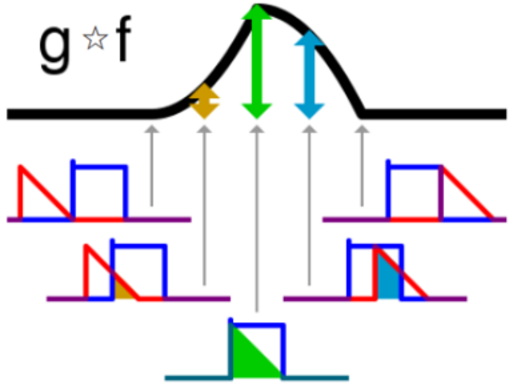
- Correlation



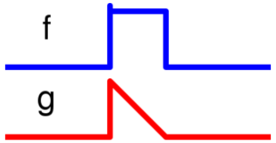


$$R(x) = f(x) \star g(x)$$

$$R(x) = \int_{-\infty}^{\infty} f(z)g(x+z)dz$$



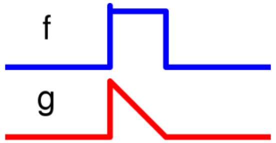
- Convolution

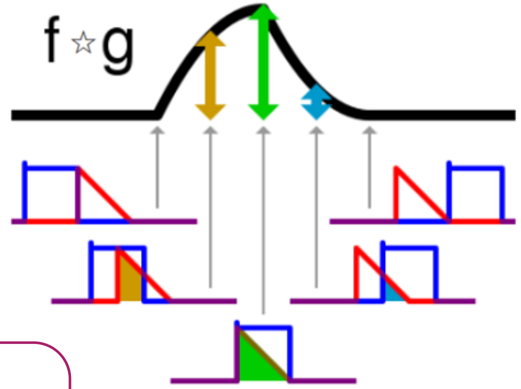


credit: wikipedia

Correlation & Convolution

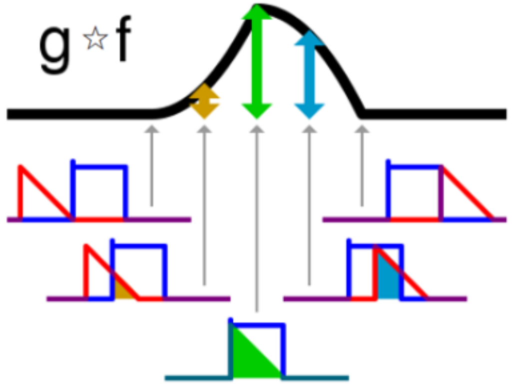
- Correlation



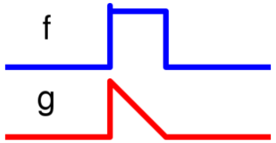


$$R(x) = f(x) \star g(x)$$

$$R(x) = \int_{-\infty}^{\infty} f(z) g(x+z) dz$$



- Convolution



$$G(x) = f(x) \ast g(x)$$

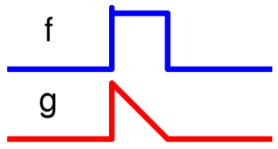
$$G(x) = \int_{-\infty}^{\infty} f(z) g(x-z) dz$$

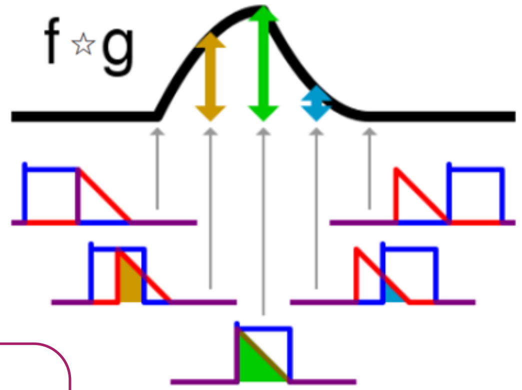
credit: wikipedia

‘ ⛤, ⊛, ✽, ✺, ✶, ✷, ✬, ✭, ✫, ☆, ✡, ✰, ✱, ❋, ✪ ’

Correlation & Convolution

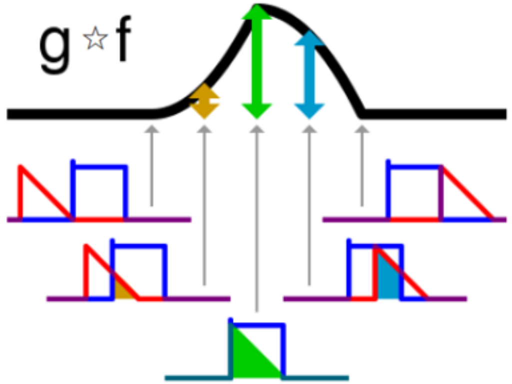
- Correlation



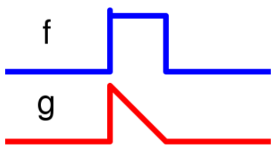


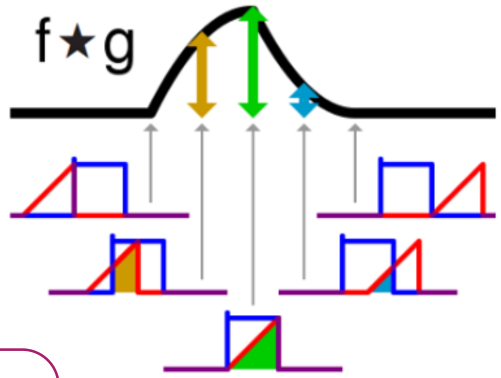
$$R(x) = f(x) \star g(x)$$

$$R(x) = \int_{-\infty}^{\infty} f(z)g(x+z)dz$$



- Convolution





$$G(x) = f(x) \star g(x)$$

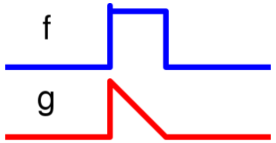
$$G(x) = \int_{-\infty}^{\infty} f(z)g(x-z)dz$$

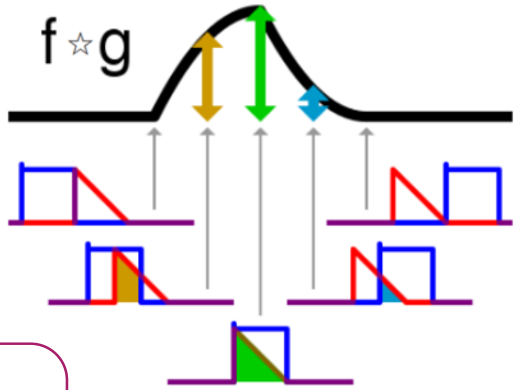
credit: wikipedia

Correlation & Convolution

- Correlation

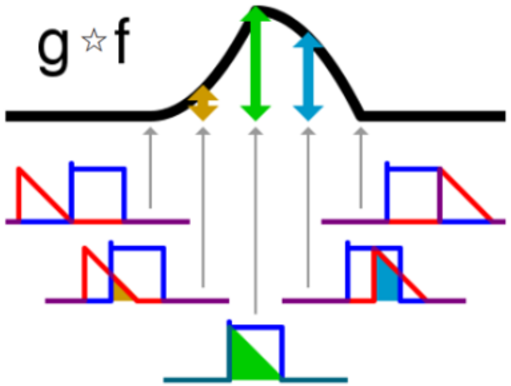
Start with the first signals last point in contact with second signal. Keep the second signal fixed and move the first signal over it. Now see the area enclosed by them and plot it.





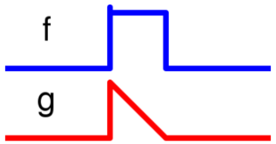
$$R(x) = f(x) \star g(x)$$

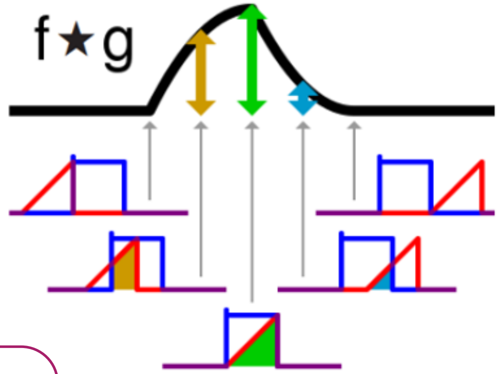
$$R(x) = \int_{-\infty}^{\infty} f(z) g(x+z) dz$$



- Convolution

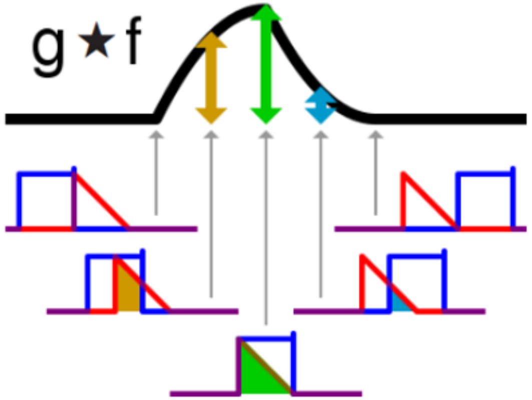
Reverse the second signal. Now place the last point of second signal in contact with the first point of first signal. Keep the first signal fixed. And keep moving the reversed second signal. Plot the area enclosed.





$$G(x) = f(x) * g(x)$$

$$G(x) = \int_{-\infty}^{\infty} f(z) g(x-z) dz$$



credit: wikipedia

Correlation & Convolution

- 2D correlation
  - cross-correlation
  - filtering algos internally use it
    - w need to be appropriately reflected before filtering

$$(w \star f)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

Correlation & Convolution

- 2D correlation
  - cross-correlation
  - filtering algos internally use it
    - w need to be appropriately reflected before filtering

$$(w \star f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

$$(w \bigstar f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

Correlation & Convolution

- 2D correlation
  - cross-correlation
  - filtering algos internally use it
    - w need to be appropriately reflected before filtering

$$(w \star f)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

W is the kernel. F is the image function

- 2D convolution
  - $w \rightarrow m \times n$
  - $a = \frac{m-1}{2}, b = \frac{n-1}{2}$
    - a, b are assumed to be odd integers
    - note the kernels do not depend on (x, y)

$$(w \bigstar f)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

Correlation & Convolution

$$(w \star f)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

$$(w \bigstar f)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

Property	**Correlation**	**Convolution**
Commutative	—	$f \bigstar g = g \bigstar f$
Associative	—	$f \bigstar (g \bigstar h) = (f \bigstar g) \bigstar h$
Distributive	$f \star (g+h) = (f \star g) + (f \star h)$	$f \bigstar (g+h) = (f \bigstar g) + (f \bigstar h)$

Filtering

- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x, y) = (w \star f)(x, y)$

Filtering

- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x, y) = (w \star f)(x, y)$

- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative

Filtering

- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x, y) = (w \star f)(x, y)$

Input

- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative

Filtering

- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x,y) = (w \star f)(x,y)$

Input

Filter-1

- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative

Filtering

- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x,y) = (w \star f)(x,y)$
- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative

Input

Filter-1

Filter-2

Filtering

- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x,y) = (w \star f)(x,y)$

- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative

Input

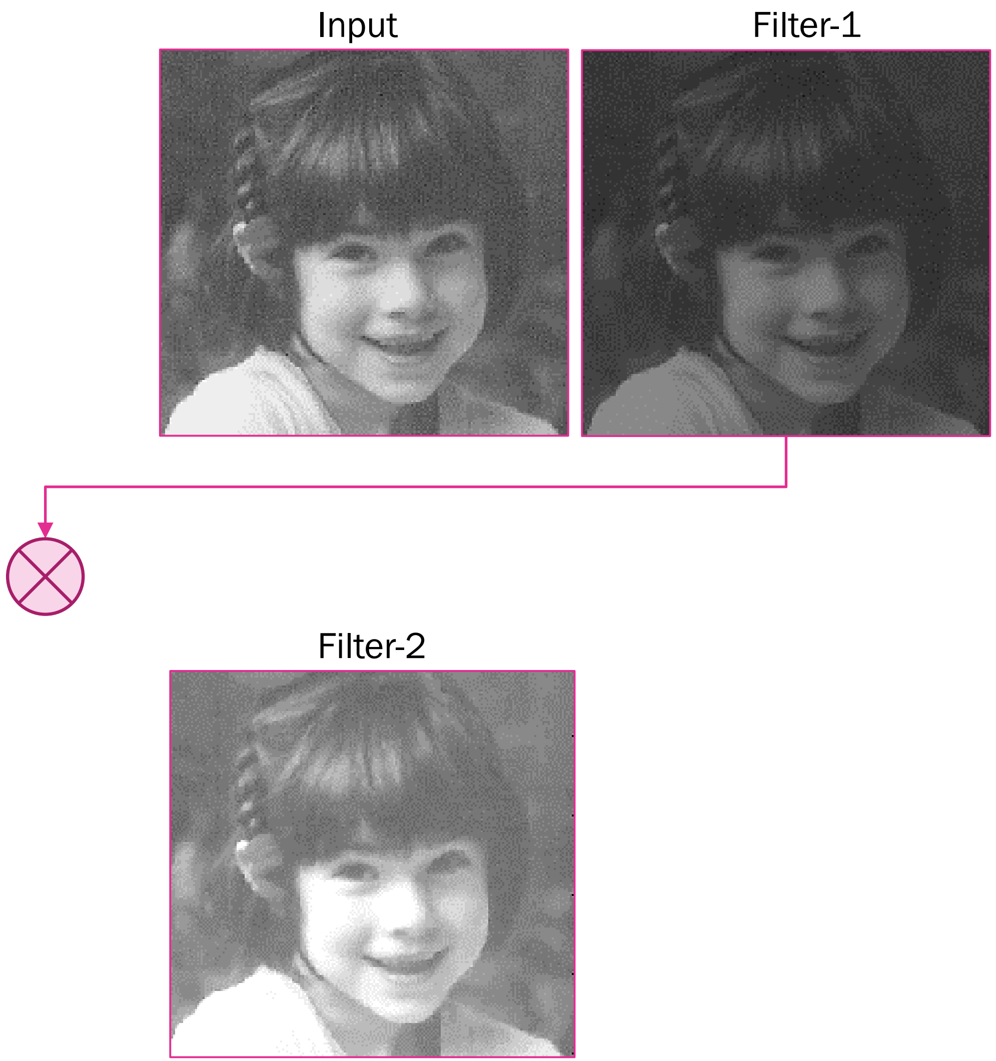
Filter-1

Filter-2

Filtering

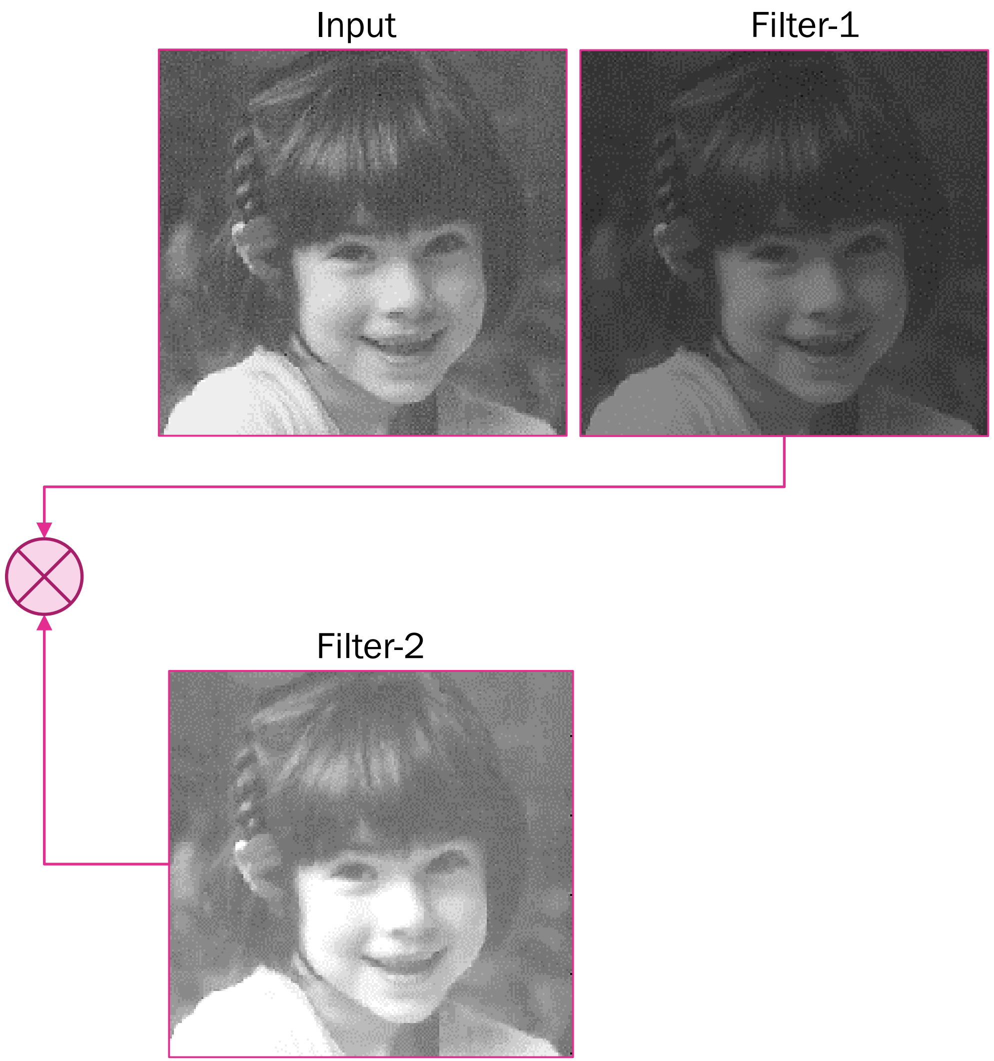
- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x,y) = (w \star f)(x,y)$

- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative



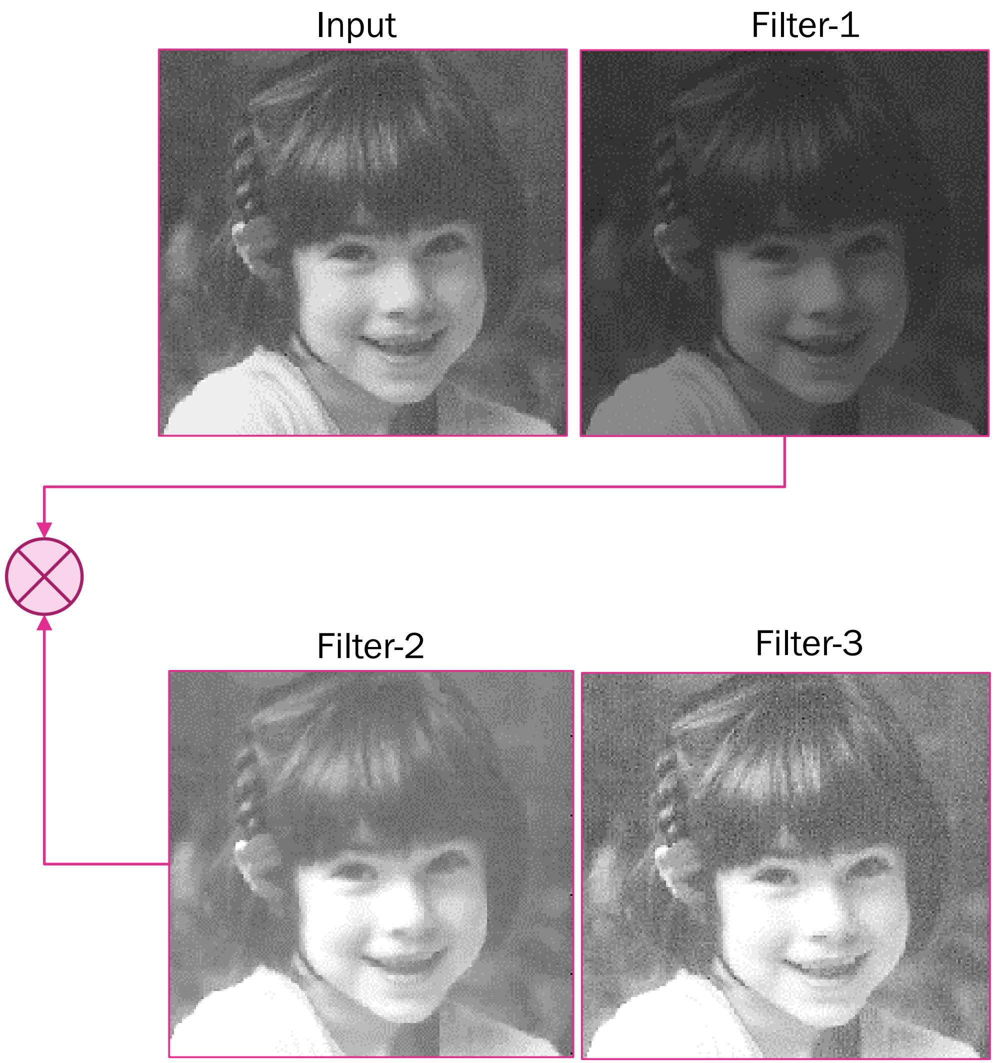
Filtering

- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x,y) = (w \star f)(x,y)$
- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative



Filtering

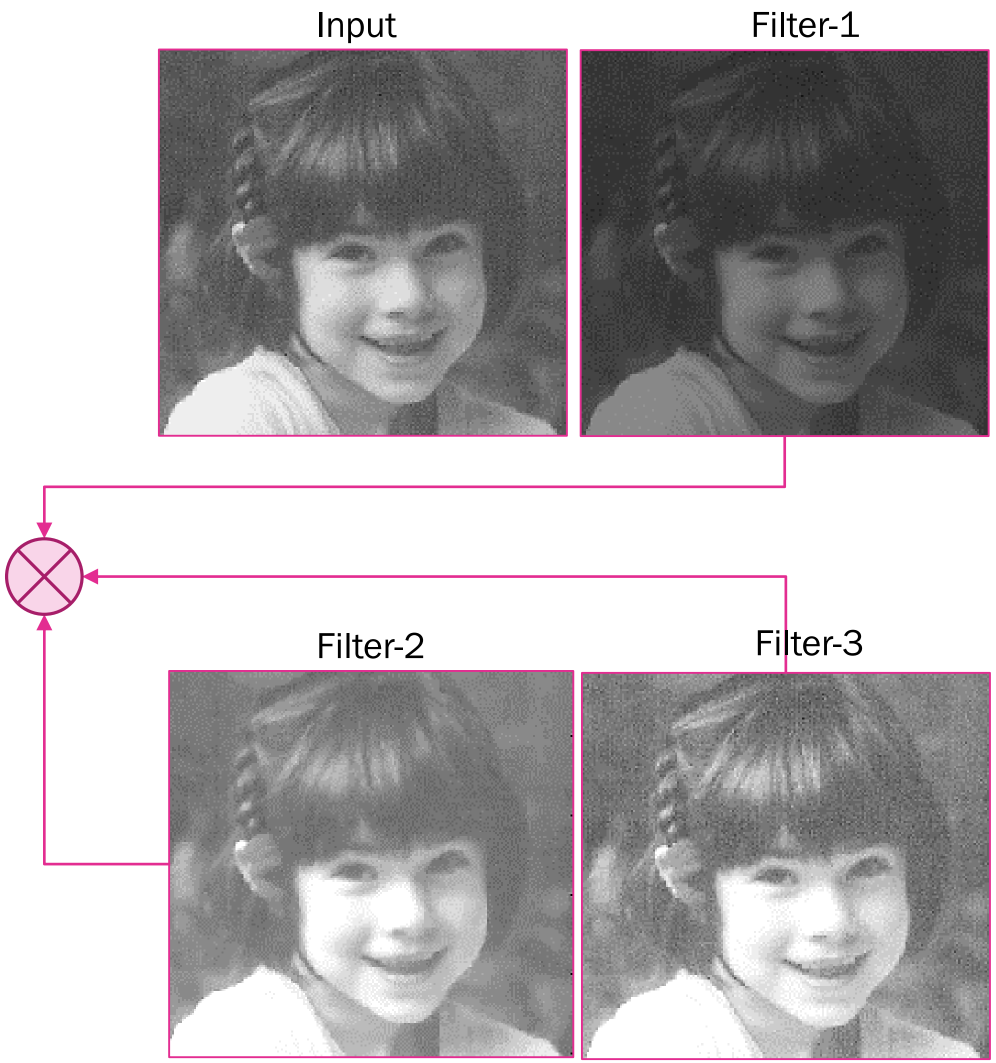
- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x,y) = (w \star f)(x,y)$
- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative



Filtering

- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x,y) = (w \star f)(x,y)$

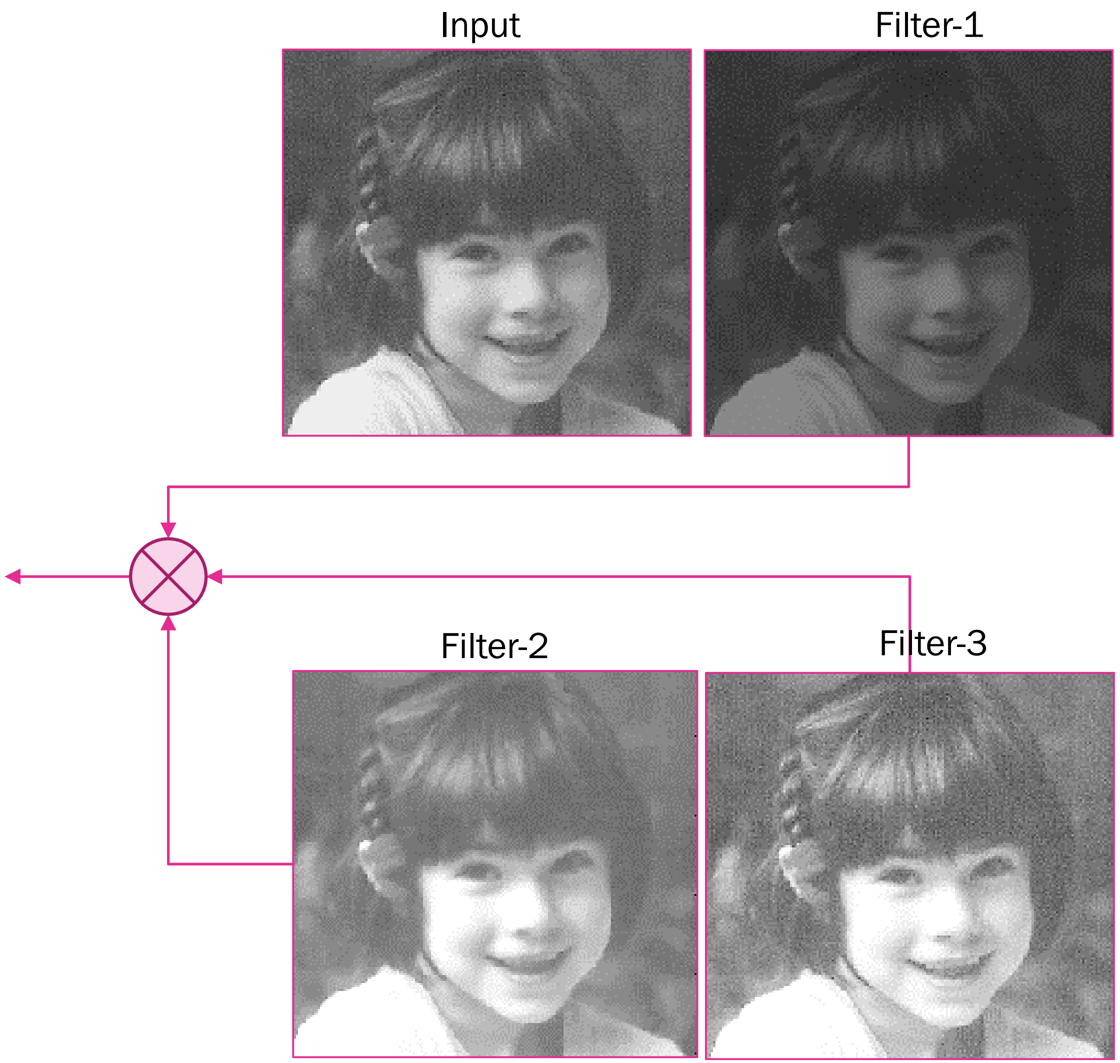
- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative



Filtering

- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x, y) = (w \star f)(x, y)$

- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative

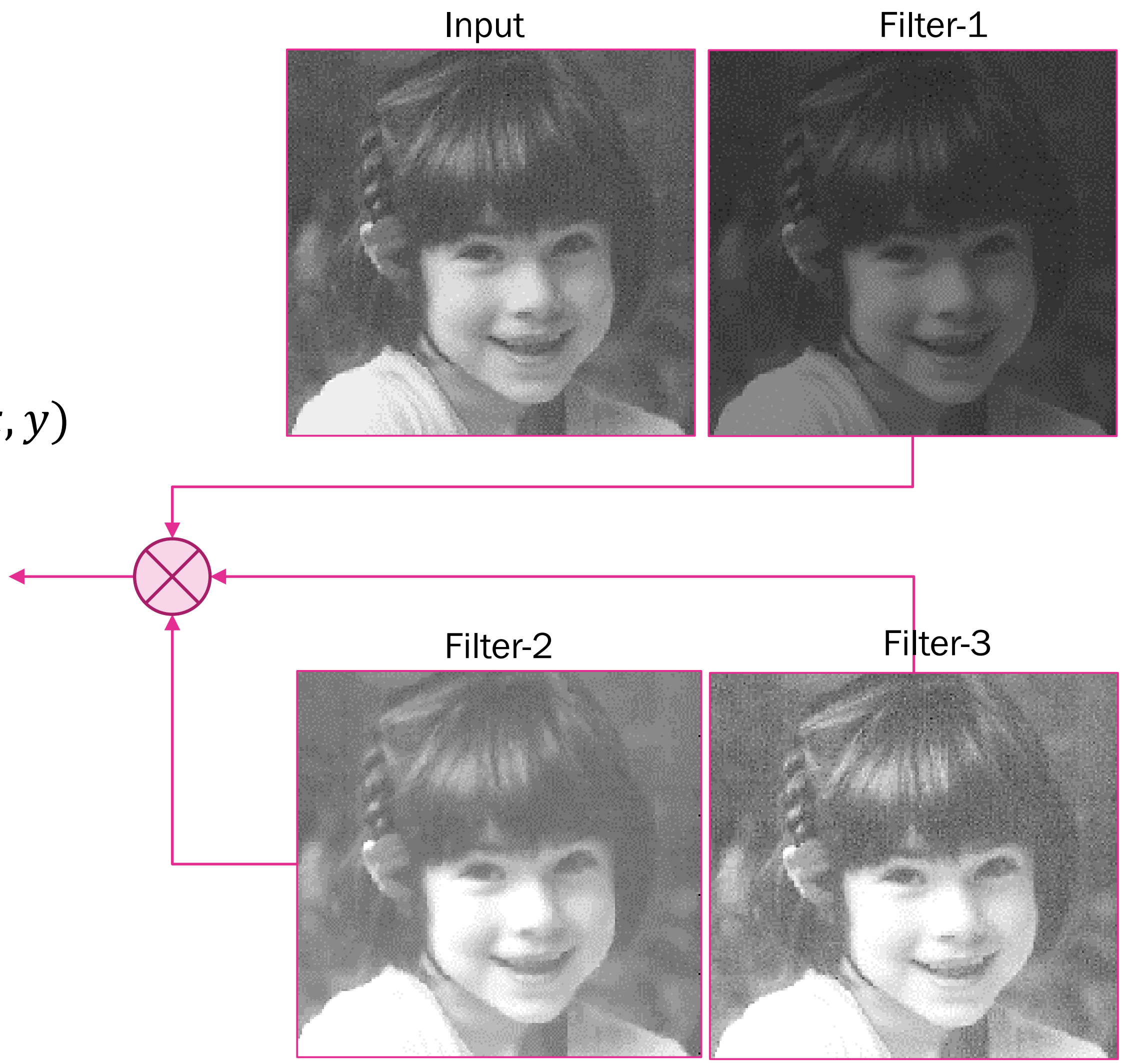


Filtering

- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x,y) = (w \star f)(x,y)$

$$g(x,y) = w_3 \star w_2 \star w_1 \star f(x,y)$$

- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative

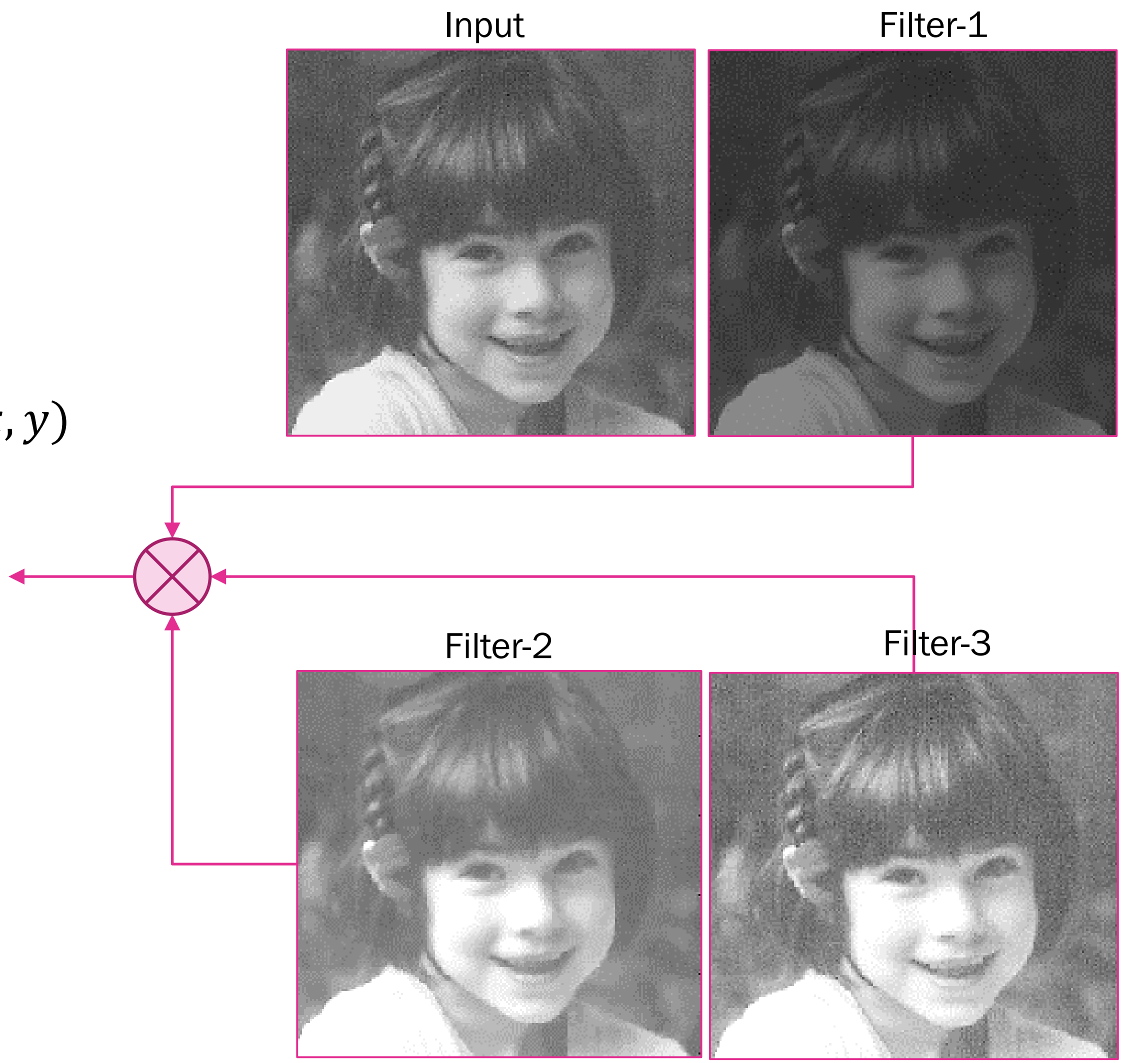


Filtering

- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x,y) = (w \star f)(x,y)$

$$g(x,y) = w_3 \star w_2 \star w_1 \star f(x,y)$$

$$(w_3 \star w_2 \star w_1) \star f(x,y)$$



- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative

Filtering

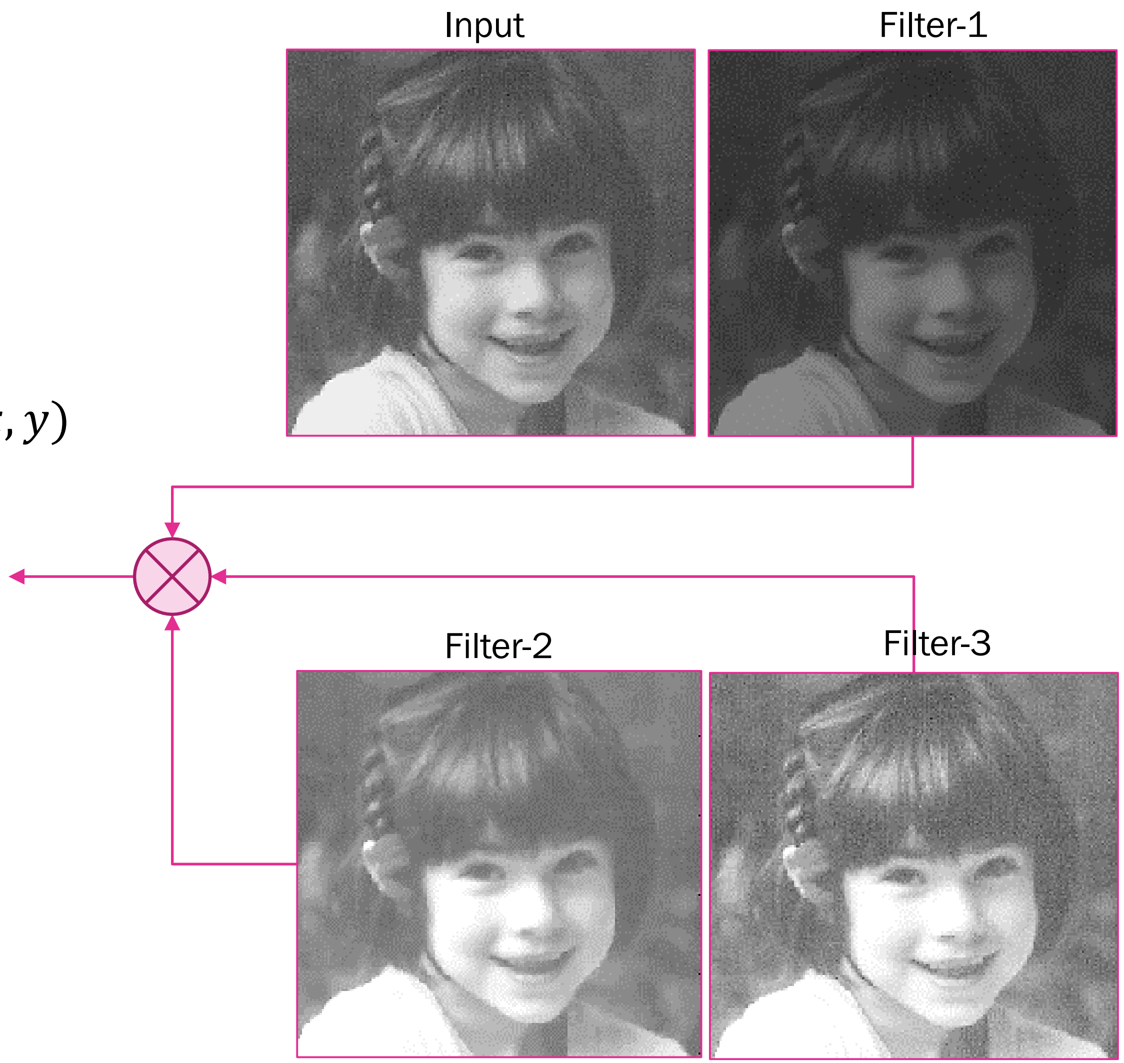
- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x,y) = (w \star f)(x,y)$

$$g(x,y) = w_3 \star w_2 \star w_1 \star f(x,y)$$

- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative

$$(w_3 \star w_2 \star w_1) \star f(x,y)$$

$$(w_1 \star w_2 \star w_3) \star f(x,y)$$



Filtering

- Image filtering
  - spatial filtering
  - convolving a kernel with an image
  - filtering: $g(x, y) = (w \star f)(x, y)$

$$g(x, y) = \ w_3 \star w_2 \star w_1 \star f(x, y)$$

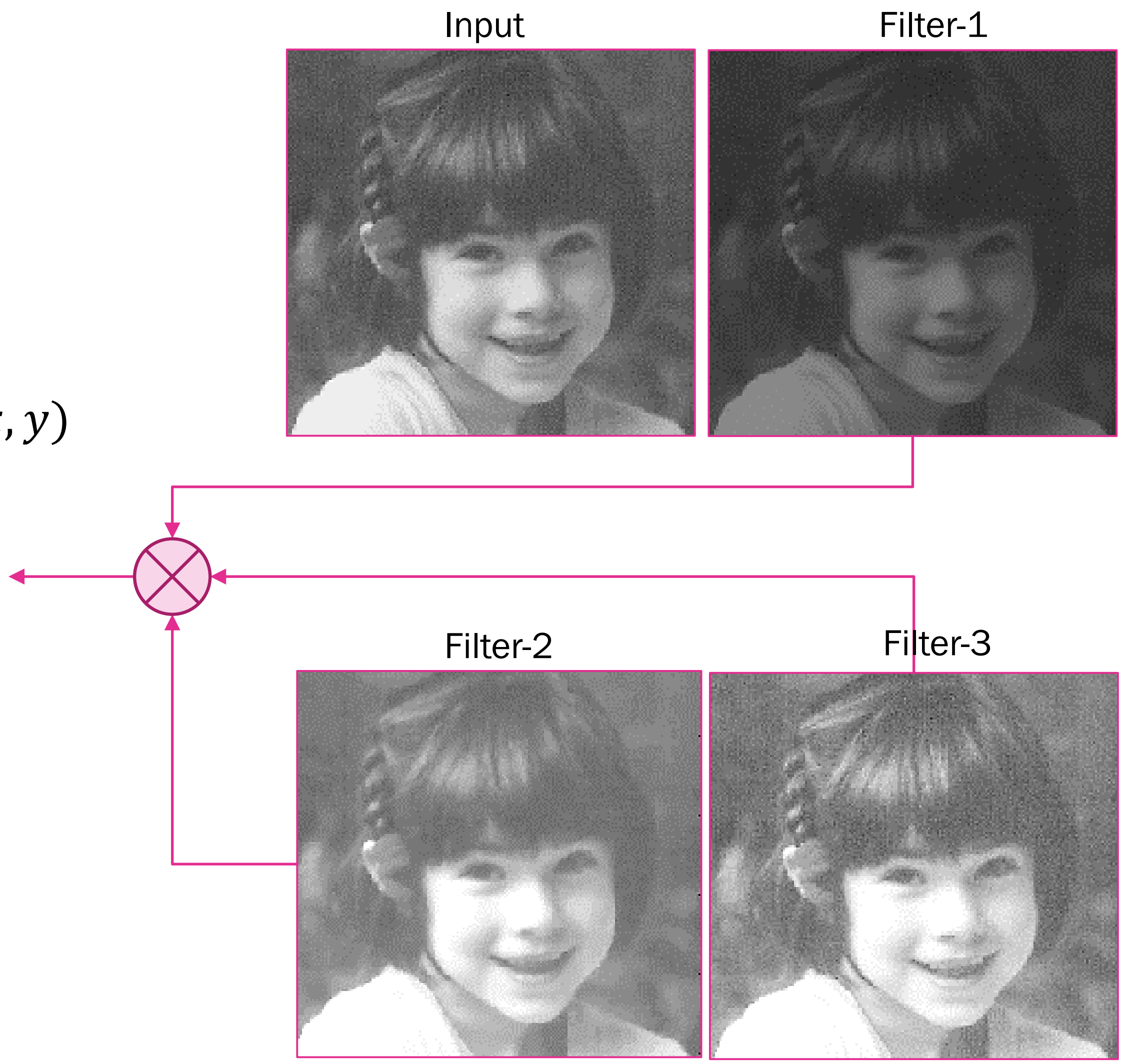
- Multistage filtering
  - filtering the filtered
  - use properties
    - commutative & associative

$$(w_3 \star w_2 \star w_1) \star f(x, y)$$

$$(w_1 \star w_2 \star w_3) \star f(x, y)$$

$$g(x, y) = \ w \star f(x, y)$$

$$w = w_1 \star w_2 \star w_3$$



Filtering

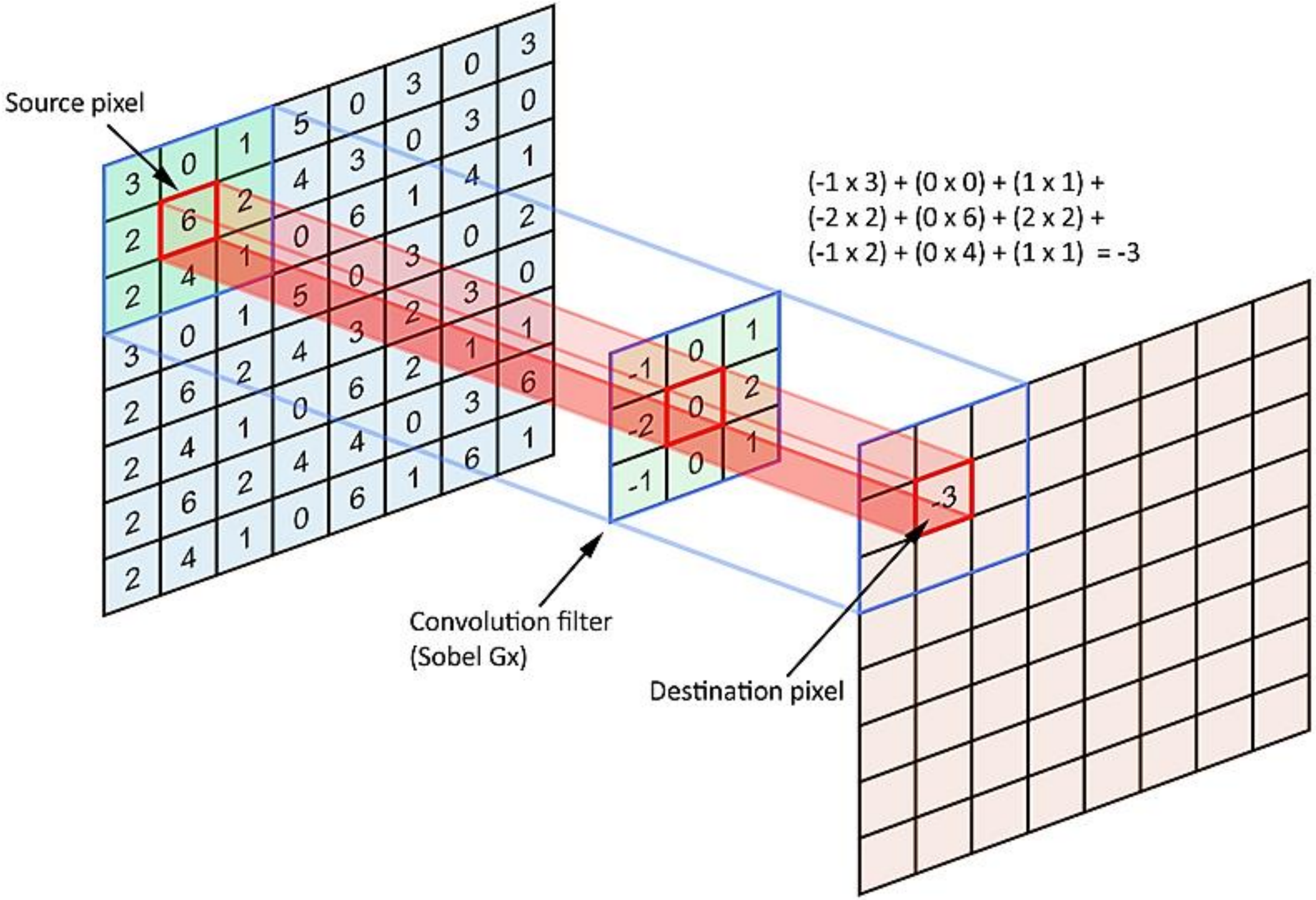
- Filter
  - kernel, mask, window, template
  - $w(i,j)$ or $k(i,j)\ \ \forall\, i,j \in N_K,\ \ K$- kernel size
    - K : determine neighbourhood of operation
    - $w(i,j)$: filter coefficients – determine nature of the filter

Filtering

- Filter
  - kernel, mask, window, template
  - $w(i,j)$ or $k(i,j)\ \ \forall\, i,j \in N_K,\ \ K$- kernel size
    - K : determine neighbourhood of operation
    - $w(i,j)$: filter coefficients – determine nature of the filter

- Nature of a filter
  - neighbour interactions
    - filter coefficients define severity of interaction
  - smoothing
  - sharpening
  - noise handling capacity

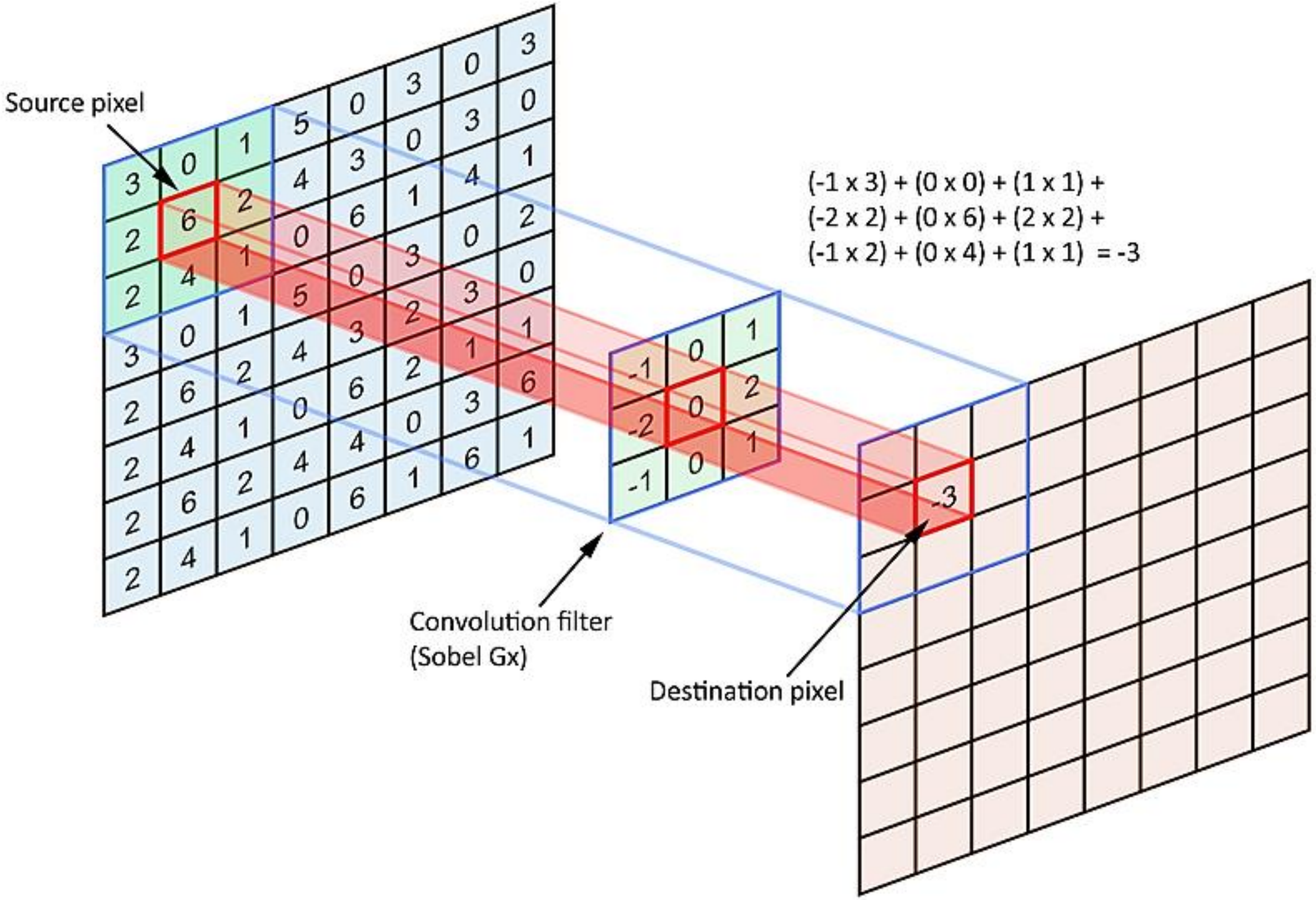
Filtering



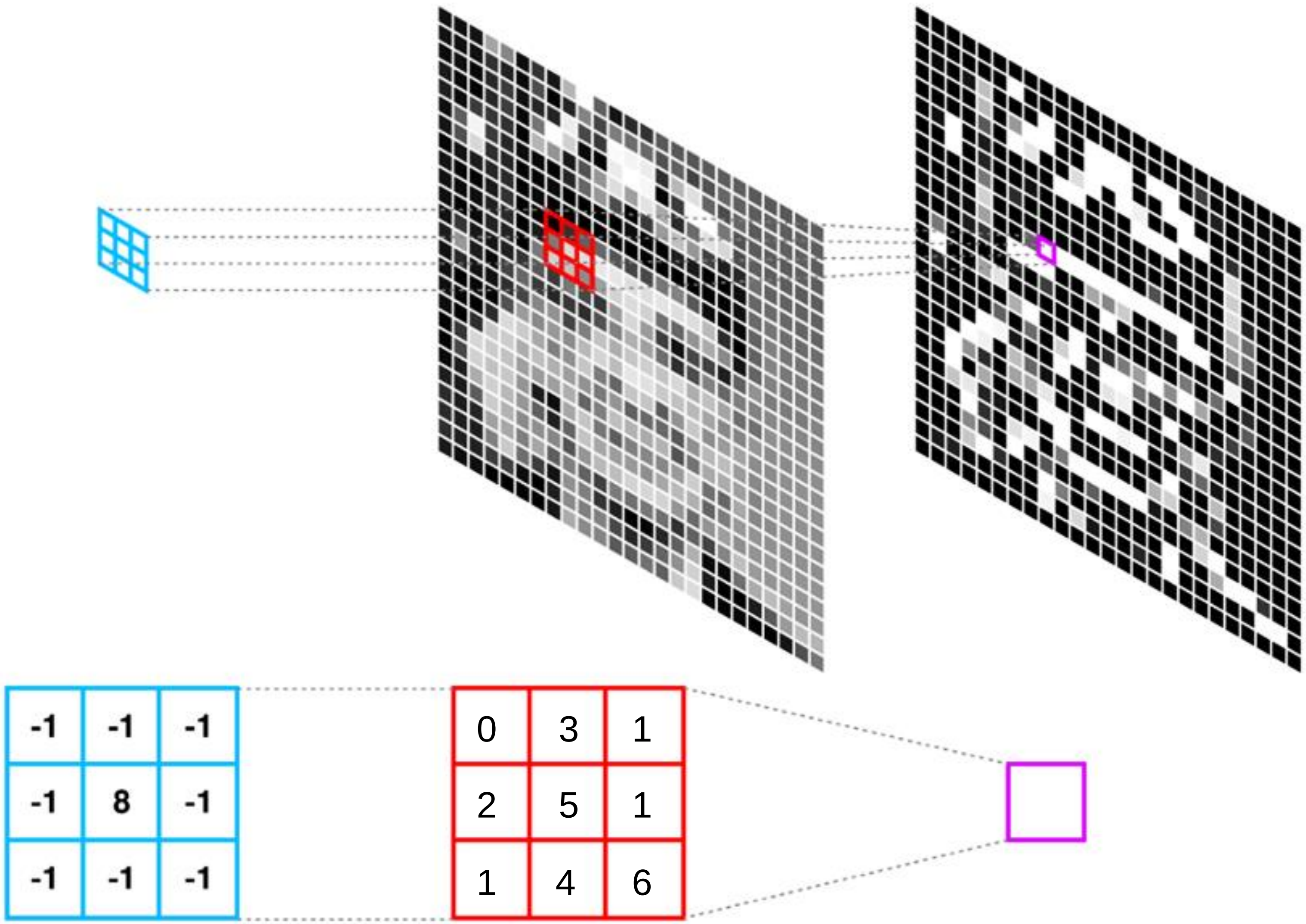
Source pixel
(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3
Convolution filter
(Sobel Gx)
Destination pixel

Filtering

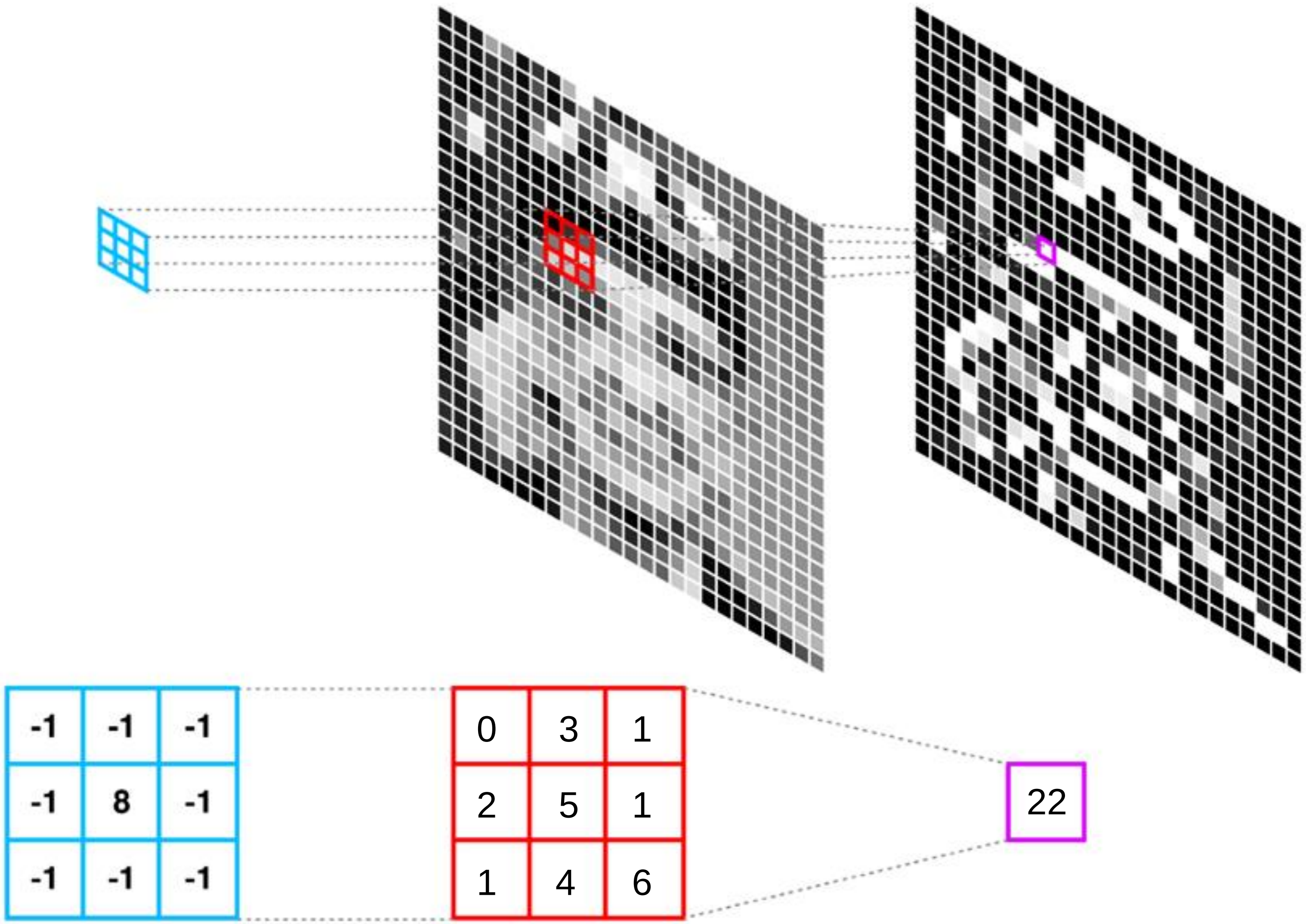
- Paddings
  - zero
  - mirror
  - replicate



Filtering



Filtering



Filtering

- Separable kernels
  - a kernel in a matrix form can be represented as outer product of two vectors
  - $w = uv^T$
    - $u \in m \times 1$
    - $v \in n \times 1$
    - sq. kernels $w = uu^T,\ w \in m \times m$

$$w = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Filtering

- Separable kernels
  - a kernel in a matrix form can be represented as outer product of two vectors
  - $w = uv^T$
    - $u \in m \times 1$
    - $v \in n \times 1$
    - sq. kernels $w = uu^T,\ w \in m \times m$

$$w = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Advantage: separable kernels
  - computationally fast
    - outer product of vectors is same as their 2D conv

Filtering

- Separable kernels
  - a kernel in a matrix form can be represented as outer product of two vectors
  - $w = uv^T$
    - $u \in m \times 1$
    - $v \in n \times 1$
    - sq. kernels $w = uu^T, \; w \in m \times m$
- Advantage: separable kernels
  - computationally fast
    - outer product of vectors is same as their 2D conv

$$w = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{c} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \mathbf{r} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Filtering

- Separable kernels
  - a kernel in a matrix form can be represented as outer product of two vectors
  - $w = uv^T$
    - $u \in m \times 1$
    - $v \in n \times 1$
    - sq. kernels $w = uu^T,\ w \in m \times m$
- Advantage: separable kernels
  - computationally fast
    - outer product of vectors is same as their 2D conv

$$w = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{c} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \mathbf{r} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\mathbf{c}\,\mathbf{r}^T = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = w$$

Filtering

- Separable kernels
  - a kernel in a matrix form can be represented as outer product of two vectors
  - $w = uv^T$
    - $u \in m \times 1$
    - $v \in n \times 1$
    - sq. kernels $w = uu^T,\ w \in m \times m$

$$w = w_1 \star w_2$$

Filtering

- Separable kernels
  - a kernel in a matrix form can be represented as outer product of two vectors
  - $w = uv^T$
    - $u \in m \times 1$
    - $v \in n \times 1$
    - sq. kernels $w = uu^T, \; w \in m \times m$

$$w = w_1 \star w_2$$

- Advantage: separable kernels
  - computationally fast
    - outer product of vectors is same as their 2D conv
    - image: $M \times N$
    -

Filtering

- Separable kernels
  - a kernel in a matrix form can be represented as outer product of two vectors
  - $w = uv^T$
    - $u \in m \times 1$
    - $v \in n \times 1$
    - sq. kernels $w = uu^T, \ w \in m \times m$

$$w = w_1 \star w_2$$

$$w \star f = (w_1 \star w_2) \star f$$

- Advantage: separable kernels
  - computationally fast
    - outer product of vectors is same as their 2D conv
    - image: $M \times N$
    -

Filtering

- Separable kernels
  - a kernel in a matrix form can be represented as outer product of two vectors
  - $w = uv^T$
    - $u \in m \times 1$
    - $v \in n \times 1$
    - sq. kernels $w = uu^T, \ w \in m \times m$
- Advantage: separable kernels
  - computationally fast
    - outer product of vectors is same as their 2D conv
    - image: $M \times N$
    -

$$w = w_1 \star w_2$$

$$w \star f = (w_1 \star w_2) \star f$$

$$= (w_2 \star w_1) \star f$$

Filtering

- Separable kernels
  - a kernel in a matrix form can be represented as outer product of two vectors
  - $w = uv^T$
    - $u \in m \times 1$
    - $v \in n \times 1$
    - sq. kernels $w = uu^T,\ w \in m \times m$

$$w = w_1 \star w_2$$

$$w \star f = (w_1 \star w_2) \star f$$

$$= (w_2 \star w_1) \star f$$

$$= w_2 \star (w_1 \star f)$$

- Advantage: separable kernels
  - computationally fast
    - outer product of vectors is same as their 2D conv
    - image: $M \times N$
    -

Filtering

- Separable kernels
  - a kernel in a matrix form can be represented as outer product of two vectors
  - $w = uv^T$
    - $u \in m \times 1$
    - $v \in n \times 1$
    - sq. kernels $w = uu^T,\ w \in m \times m$

- Advantage: separable kernels
  - computationally fast
    - outer product of vectors is same as their 2D conv
    - image: $M \times N$
    -

$$w = w_1 \star w_2$$

$$w \star f = (w_1 \star w_2) \star f$$

$$= (w_2 \star w_1) \star f$$

$$= w_2 \star (w_1 \star f)$$

$$= (w_1 \star f) \star w_2$$

Filtering

- Separable kernels
  - a kernel in a matrix form can be represented as outer product of two vectors
  - $w = uv^T$
    - $u \in m \times 1$
    - $v \in n \times 1$
    - sq. kernels $w = uu^T, \; w \in m \times m$
- Advantage: separable kernels
  - computationally fast
    - outer product of vectors is same as their 2D conv
    - image: $M \times N$
    - advantage factor $= \dfrac{mn}{m+n}$

SEE NOTES

$$w = w_1 \star w_2$$

$$w \star f = (w_1 \star w_2) \star f$$

$$= (w_2 \star w_1) \star f$$

$$= w_2 \star (w_1 \star f)$$

$$= (w_1 \star f) \star w_2$$

Filtering

- Box filter
  - smoothing filter
  - lowpass filter
  - averaging filter

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Filtering

- Box filter
  - smoothing filter
  - lowpass filter
  - averaging filter

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

- Use cases
  - random noise reduction
    - reducing sharp transitions in intensity
    - favours blurring along perpendicular directions
  - reduce aliasing
    - smoothing prior to resampling
  - reduce quantization noise
    - reduce false contours of intensities
  - essential in composite filtering
    - multistage filters

Filtering

- Box filter
  - smoothing filter
  - lowpass filter
  - averaging filter

- Use cases
  - random noise reduction
    - reducing sharp transitions in intensity
    - favours blurring along perpendicular directions
  - reduce aliasing
    - smoothing prior to resampling
  - reduce quantization noise
    - reduce false contours of intensities
  - essential in composite filtering
    - multistage filters

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Filtering

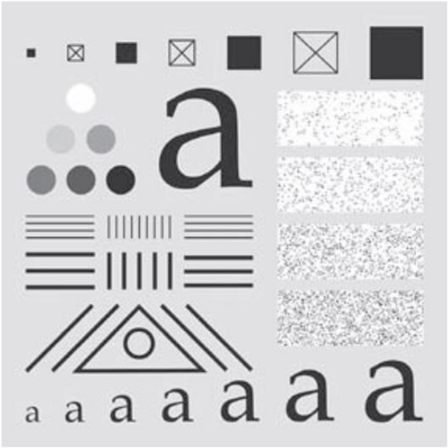
- Box filter
  - smoothing filter
  - lowpass filter
  - averaging filter

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Filtering

- Box filter
  - smoothing filter
  - lowpass filter
  - averaging filter

input

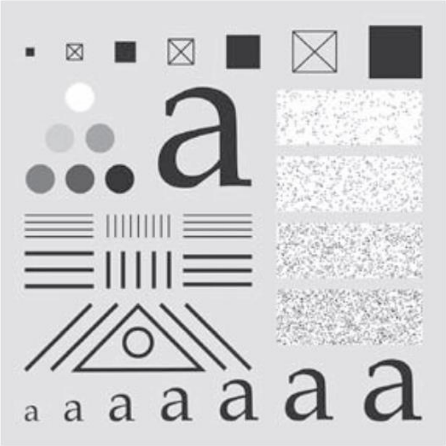


$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Filtering

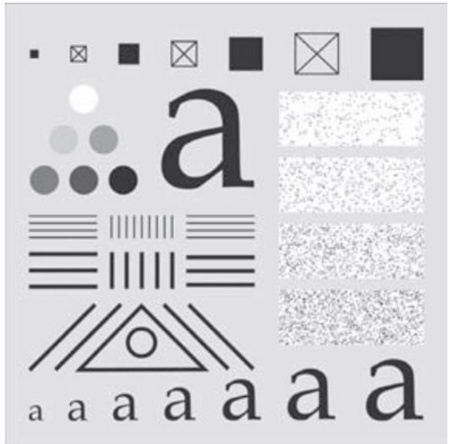
- Box filter
  - smoothing filter
  - lowpass filter
  - averaging filter

input



$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

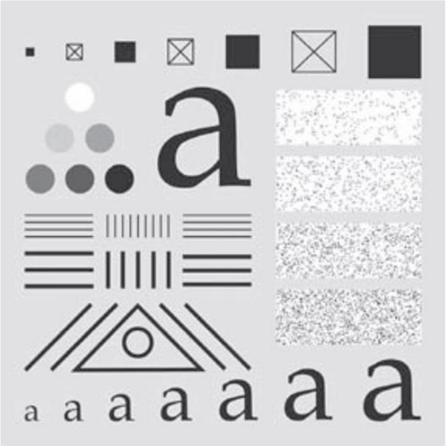
m=3



Filtering

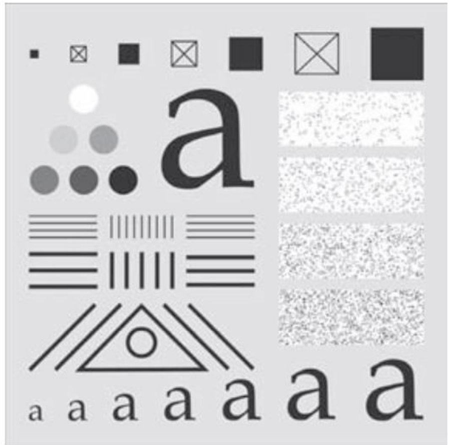
- Box filter
  - smoothing filter
  - lowpass filter
  - averaging filter

input

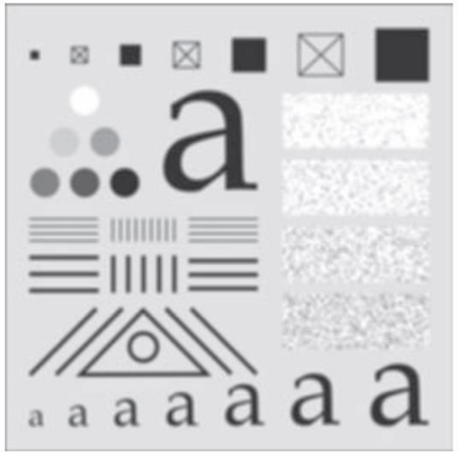


$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

m=3



m=11

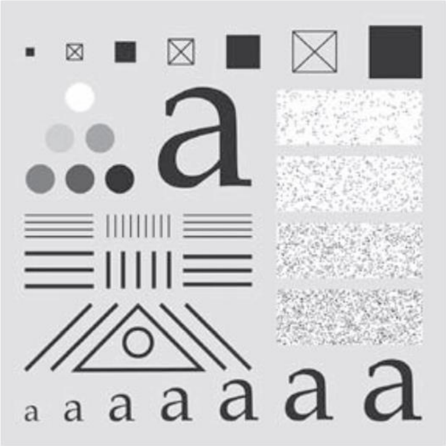


Filtering

- Box filter
  - smoothing filter
  - lowpass filter
  - averaging filter

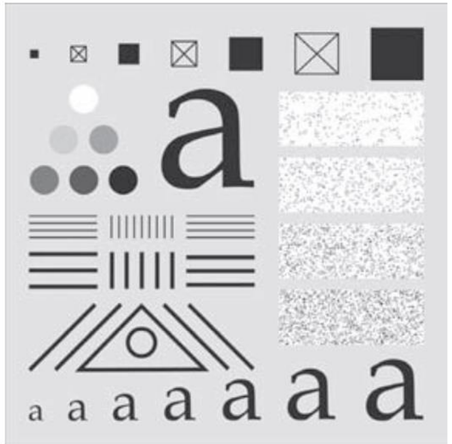
Doubt - Here m represents Kernel size (MAY BE?)

input

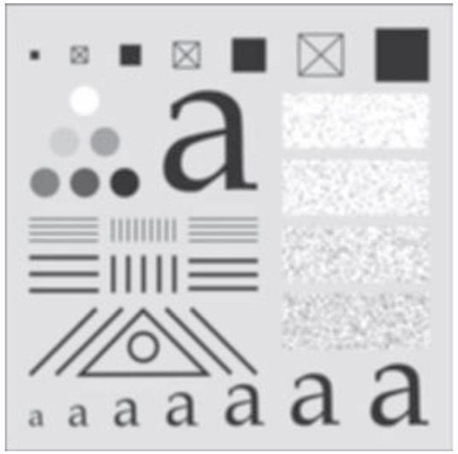


$$\frac{1}{9}\times\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

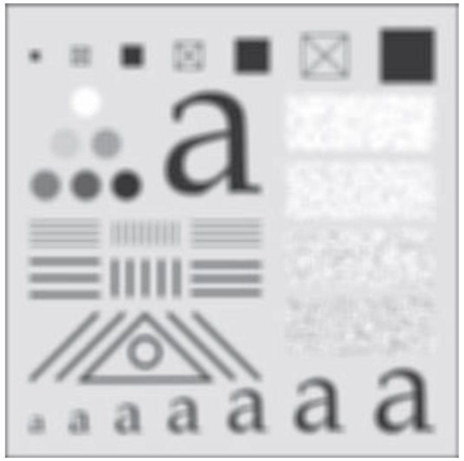
m=3



m=11



m=21



Filtering

- Gaussian filter
  - smoothing filter
  - defocused lens approximators
  - isotropic
    - response is independent of orientation
    - circularly symmetric

$$w(s,t) = G(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}}$$

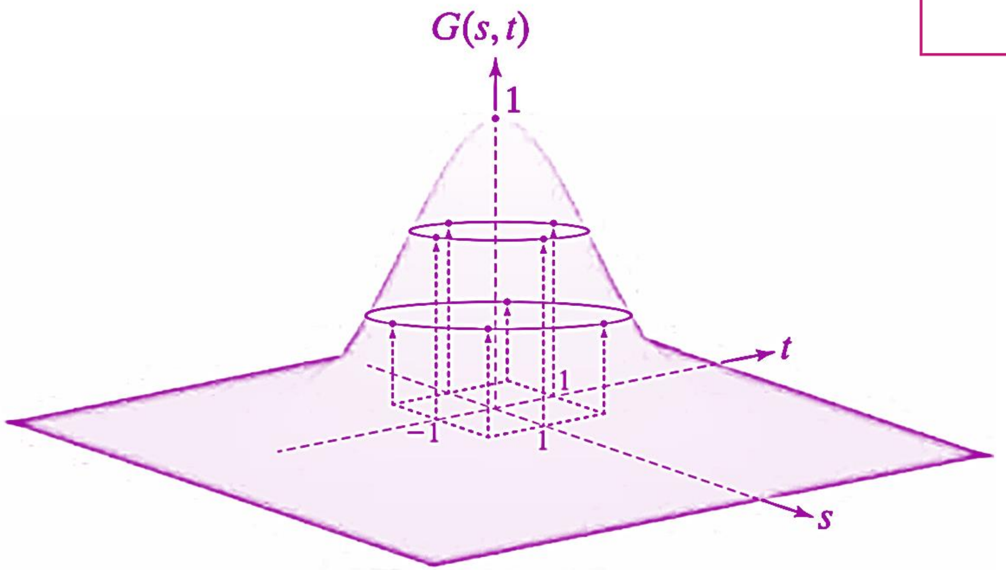
$$\frac{1}{4.8976} \times \begin{array}{|c|c|c|} \hline 0.3679 & 0.6065 & 0.3679 \\ \hline 0.6065 & 1.0000 & 0.6065 \\ \hline 0.3679 & 0.6065 & 0.3679 \\ \hline \end{array}$$

Filtering

- Gaussian filter
  - smoothing filter
  - defocused lens approximators
  - isotropic
    - response is independent of orientation
    - circularly symmetric

$$w(s,t) = G(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}}$$

$$\frac{1}{4.8976} \times \begin{bmatrix} 0.3679 & 0.6065 & 0.3679 \\ 0.6065 & 1.0000 & 0.6065 \\ 0.3679 & 0.6065 & 0.3679 \end{bmatrix}$$



Filtering

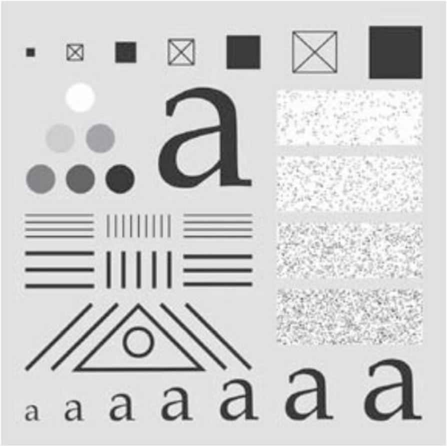
- Gaussian filter
  - smoothing filter
  - defocused lens approximators
  - isotropic
    - response is independent of orientation
    - circularly symmetric

$$\frac{1}{4.8976} \times \begin{array}{|c|c|c|} \hline 0.3679 & 0.6065 & 0.3679 \\ \hline 0.6065 & 1.0000 & 0.6065 \\ \hline 0.3679 & 0.6065 & 0.3679 \\ \hline \end{array}$$

Filtering

- Gaussian filter
  - smoothing filter
  - defocused lens approximators
  - isotropic
    - response is independent of orientation
    - circularly symmetric

input

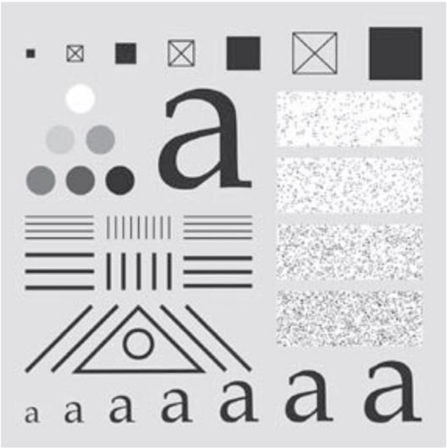


$$\frac{1}{4.8976} \times \begin{array}{|c|c|c|}\hline 0.3679 & 0.6065 & 0.3679 \\ \hline 0.6065 & 1.0000 & 0.6065 \\ \hline 0.3679 & 0.6065 & 0.3679 \\ \hline \end{array}$$

Filtering

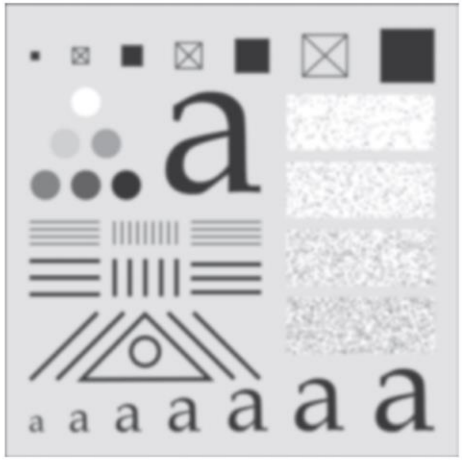
- Gaussian filter
  - smoothing filter
  - defocused lens approximators
  - isotropic
    - response is independent of orientation
    - circularly symmetric

input



$$\frac{1}{4.8976} \times \begin{array}{|c|c|c|}\hline 0.3679 & 0.6065 & 0.3679 \\ \hline 0.6065 & 1.0000 & 0.6065 \\ \hline 0.3679 & 0.6065 & 0.3679 \\ \hline \end{array}$$

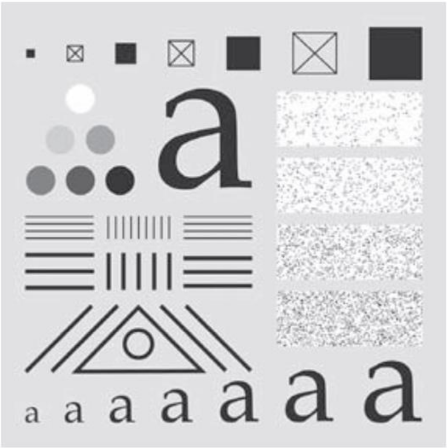
m=21 σ=3.5 Gauss



Filtering

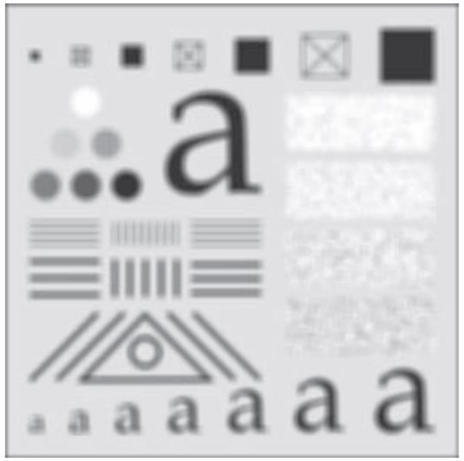
- Gaussian filter
  - smoothing filter
  - defocused lens approximators
  - isotropic
    - response is independent of orientation
    - circularly symmetric

input

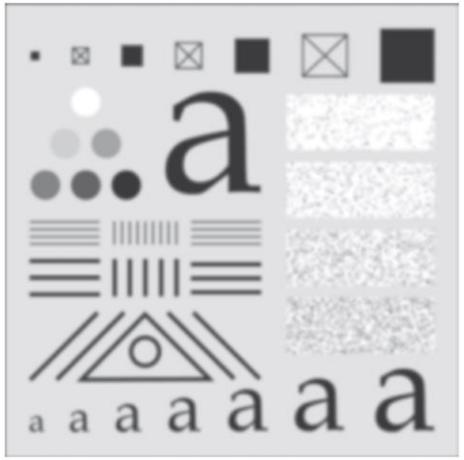


$$\frac{1}{4.8976} \times \begin{array}{|c|c|c|} \hline 0.3679 & 0.6065 & 0.3679 \\ \hline 0.6065 & 1.0000 & 0.6065 \\ \hline 0.3679 & 0.6065 & 0.3679 \\ \hline \end{array}$$

m=21 box



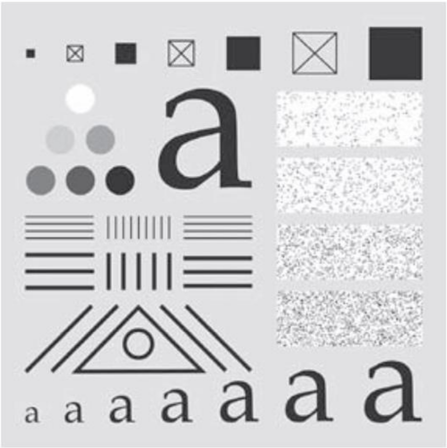
m=21 σ=3.5 Gauss



Filtering

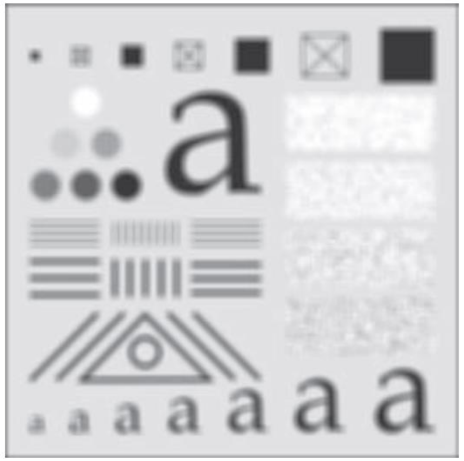
- Gaussian filter
  - smoothing filter
  - defocused lens approximators
  - isotropic
    - response is independent of orientation
    - circularly symmetric

input

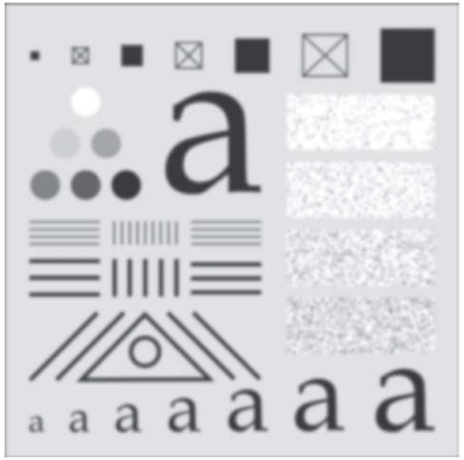


$$\frac{1}{4.8976} \times \begin{array}{|c|c|c|} \hline 0.3679 & 0.6065 & 0.3679 \\ \hline 0.6065 & 1.0000 & 0.6065 \\ \hline 0.3679 & 0.6065 & 0.3679 \\ \hline \end{array}$$

m=21 box



m=21 σ=3.5 Gauss

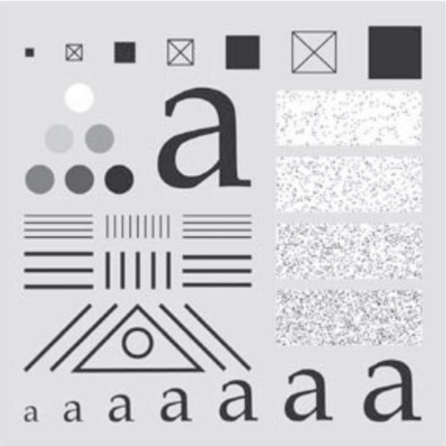


m=43 σ=7 Gauss

Filtering

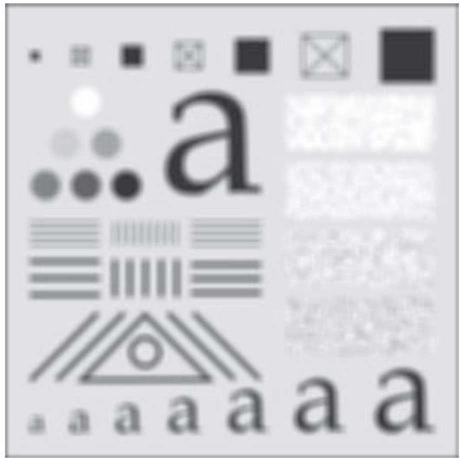
- Gaussian filter
  - smoothing filter
  - defocused lens approximators
  - isotropic
    - response is independent of orientation
    - circularly symmetric

input

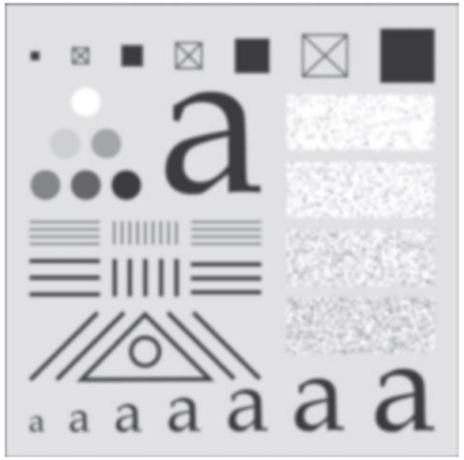


$$\frac{1}{4.8976} \times \begin{bmatrix} 0.3679 & 0.6065 & 0.3679 \\ 0.6065 & 1.0000 & 0.6065 \\ 0.3679 & 0.6065 & 0.3679 \end{bmatrix}$$

m=21 box

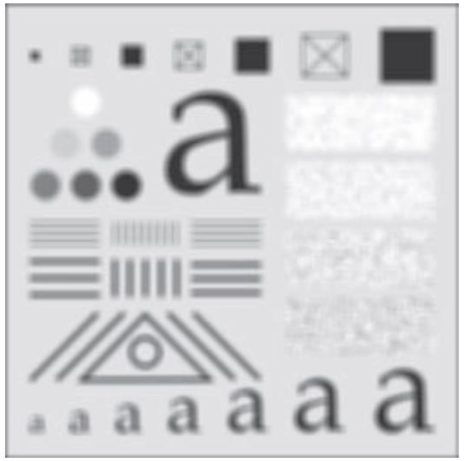


m=21 σ=3.5 Gauss



m=43 σ=7 Gauss

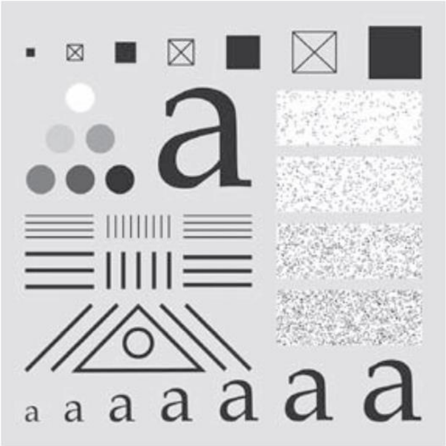
m=21 box



Filtering

- Gaussian filter
  - smoothing filter
  - defocused lens approximators
  - isotropic
    - response is independent of orientation
    - circularly symmetric

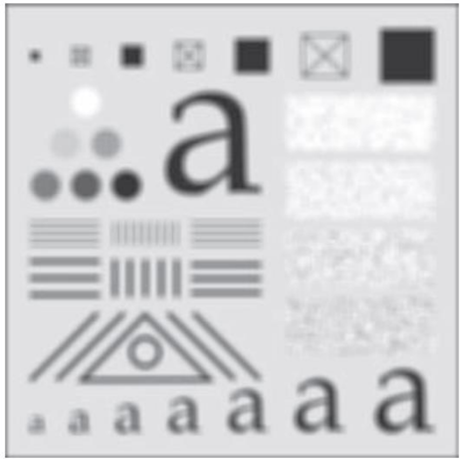
input



$$\frac{1}{4.8976} \times \begin{bmatrix} 0.3679 & 0.6065 & 0.3679 \\ 0.6065 & 1.0000 & 0.6065 \\ 0.3679 & 0.6065 & 0.3679 \end{bmatrix}$$

m=43 σ=7 Gauss

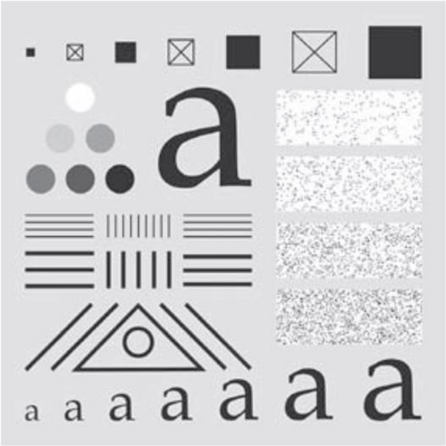
m=21 box



Filtering

- Gaussian filter
  - smoothing filter
  - defocused lens approximators
  - isotropic
    - response is independent of orientation
    - circularly symmetric

input

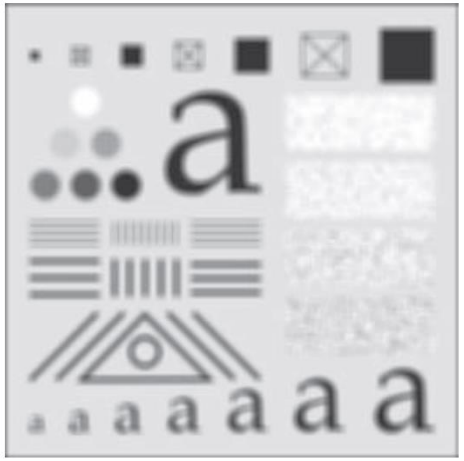


$$\frac{1}{4.8976} \times \begin{array}{|c|c|c|}\hline 0.3679 & 0.6065 & 0.3679 \\ \hline 0.6065 & 1.0000 & 0.6065 \\ \hline 0.3679 & 0.6065 & 0.3679 \\ \hline \end{array}$$

m=85 σ=7 Gauss

m=43 σ=7 Gauss

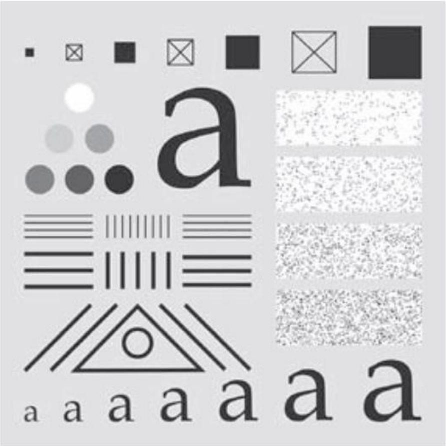
m=21 box



Filtering

- Gaussian filter
  - smoothing filter
  - defocused lens approximators
  - isotropic
    - response is independent of orientation
    - circularly symmetric

input



$$\frac{1}{4.8976} \times$$

0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

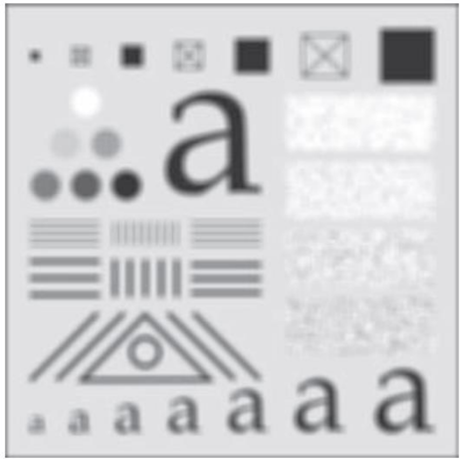
SEE Notes and slides together

difference $m_{85} - m_{43}$

m=85 σ=7 Gauss

m=43 σ=7 Gauss

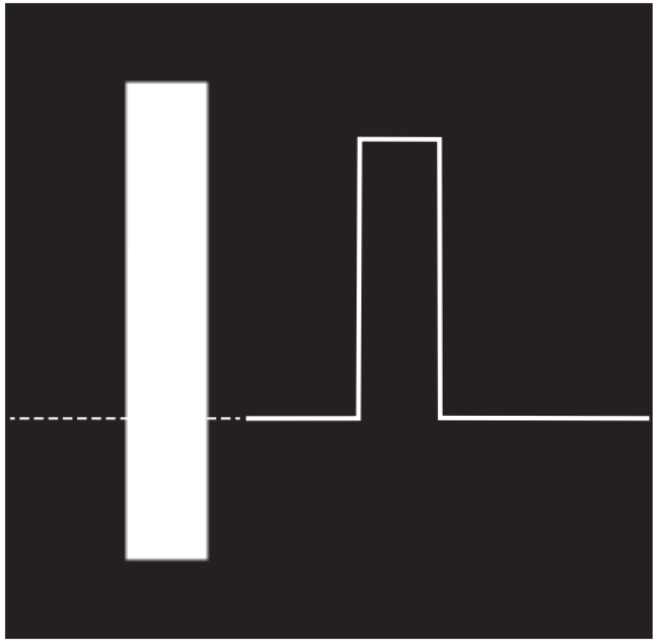
m=21 box



This is the difference image which is black => no difference in img 1 and img 2

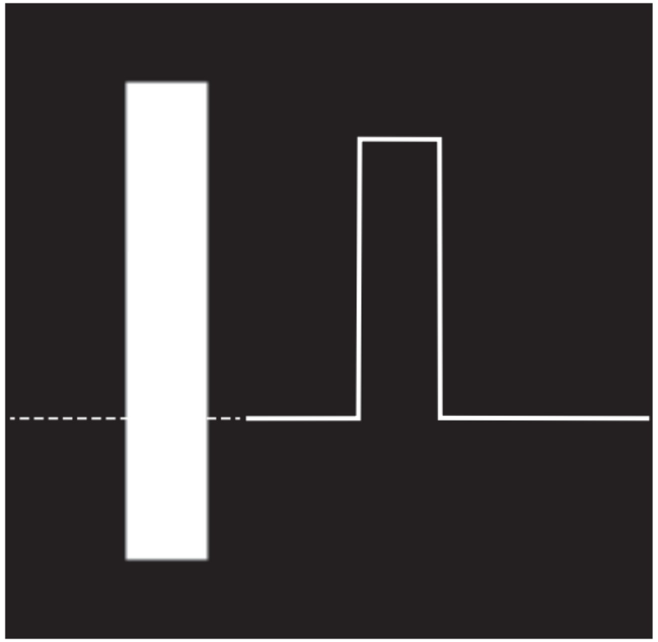
Filtering

- Box vs Gaussian
  - blur profile
  - blurred rects having same shape

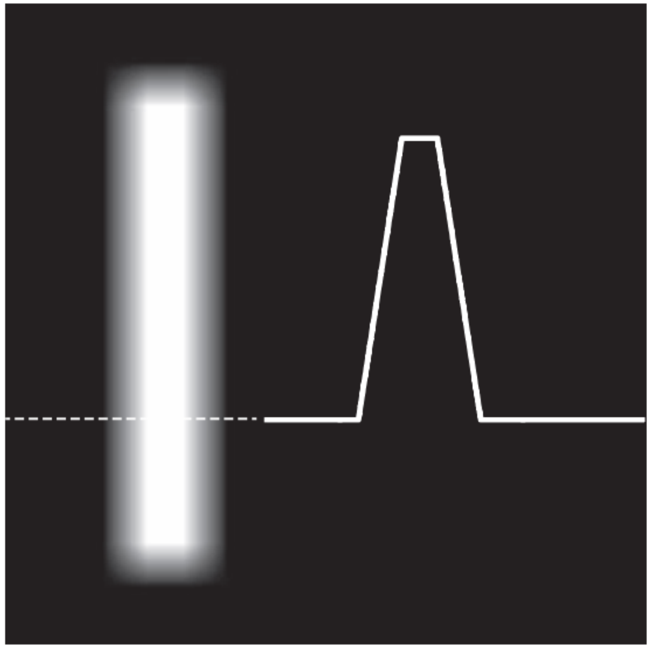


Filtering

- Box vs Gaussian
  - blur profile
  - blurred rects having same shape

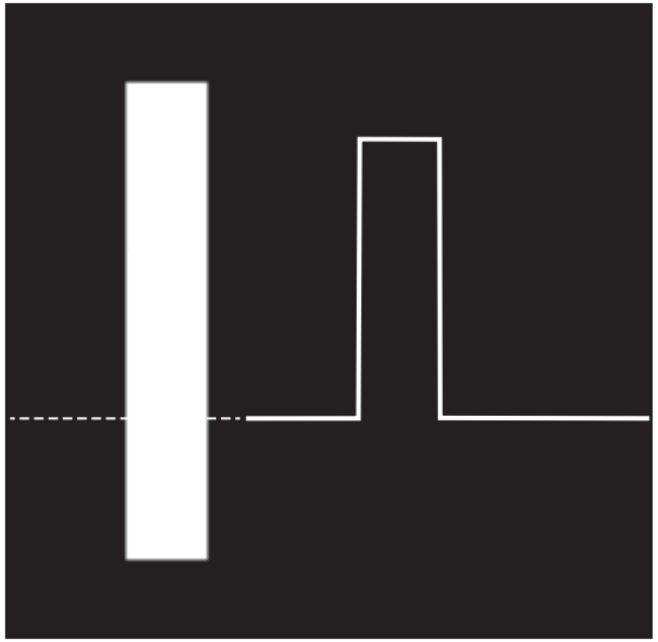


m=71 box

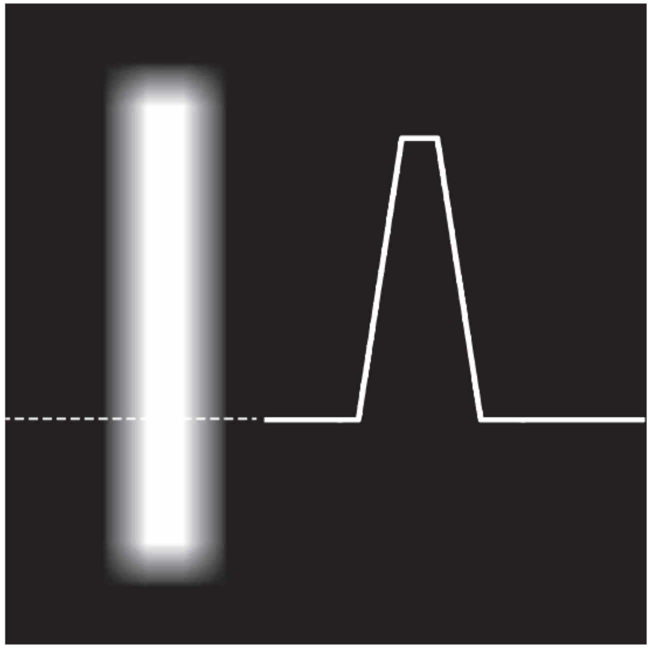


Filtering

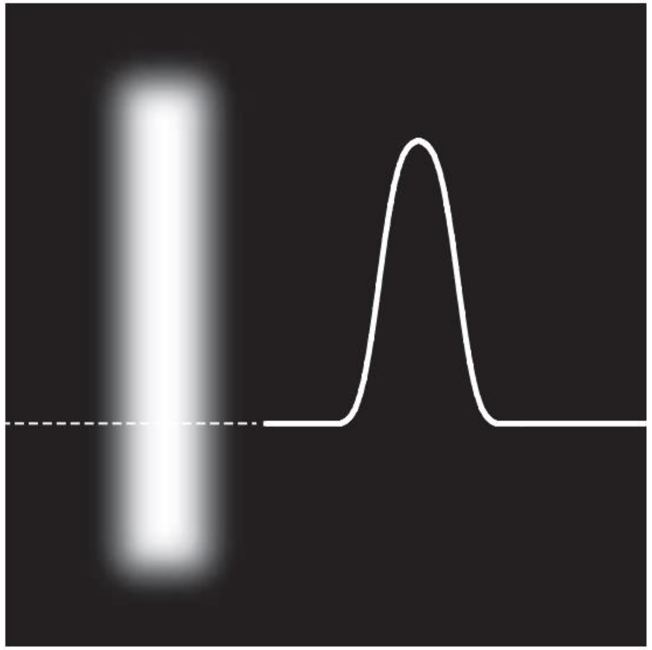
- Box vs Gaussian
  - blur profile
  - blurred rects having same shape



m=71 box



m=151 σ=25 Gauss

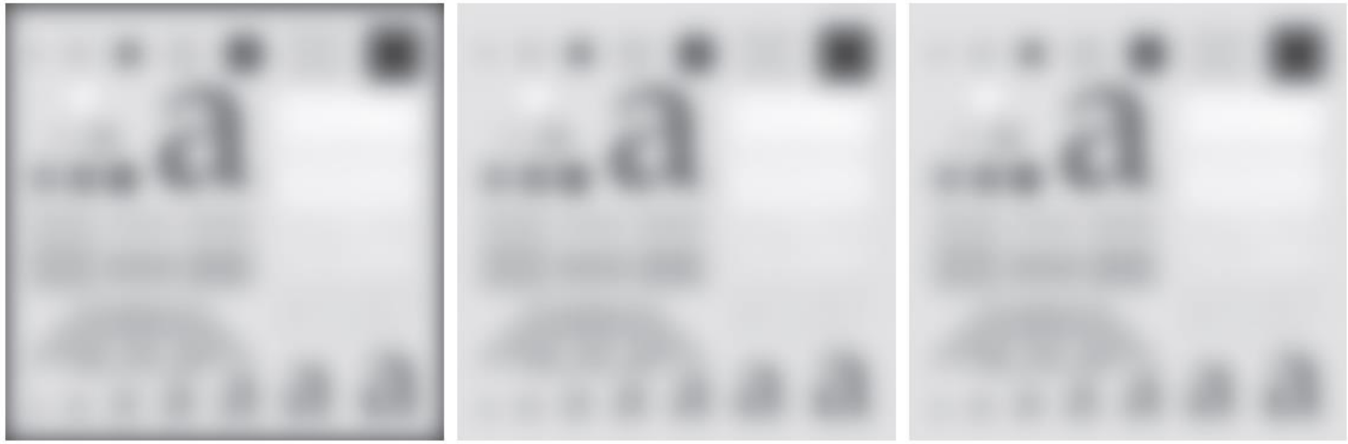


Filtering

- Padding effects

m=187 σ=31 Gauss

image 1024x1024



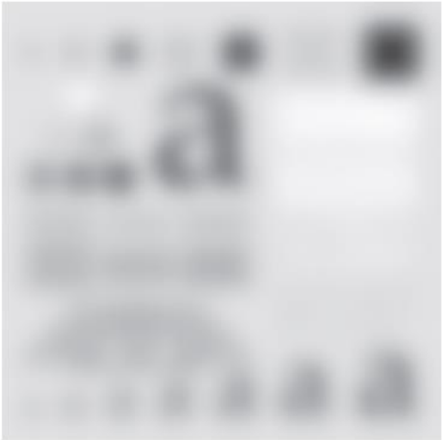
Filtering

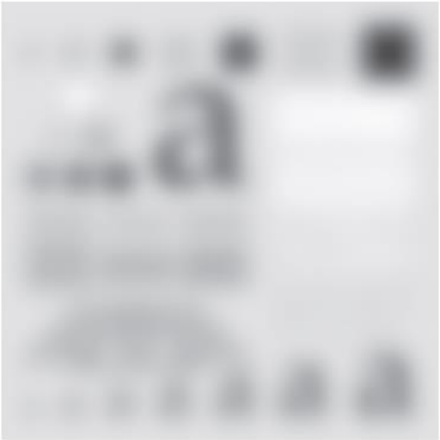
- Padding effects

m=187 σ=31 Gauss

image 1024x1024

zero | mirror | replicate





Filtering

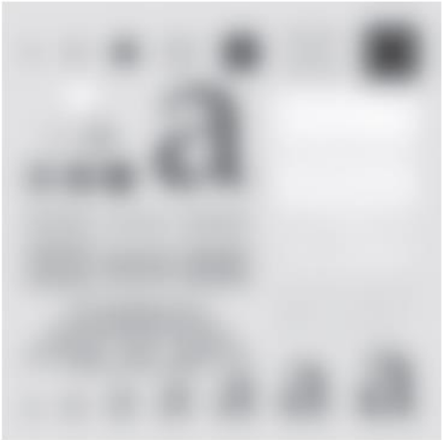
- Padding effects

m=187 σ=31 Gauss

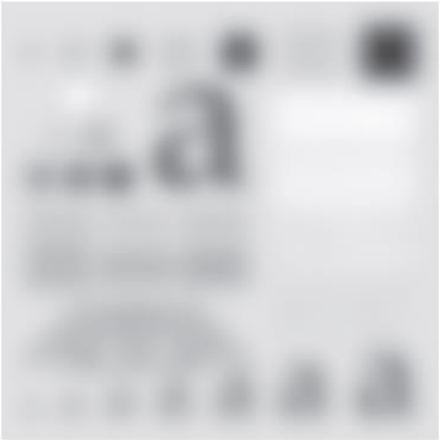
image 1024x1024

zero

mirror



replicate



- Relative size effect

m=187 σ=31 Gauss

image 4096x4096

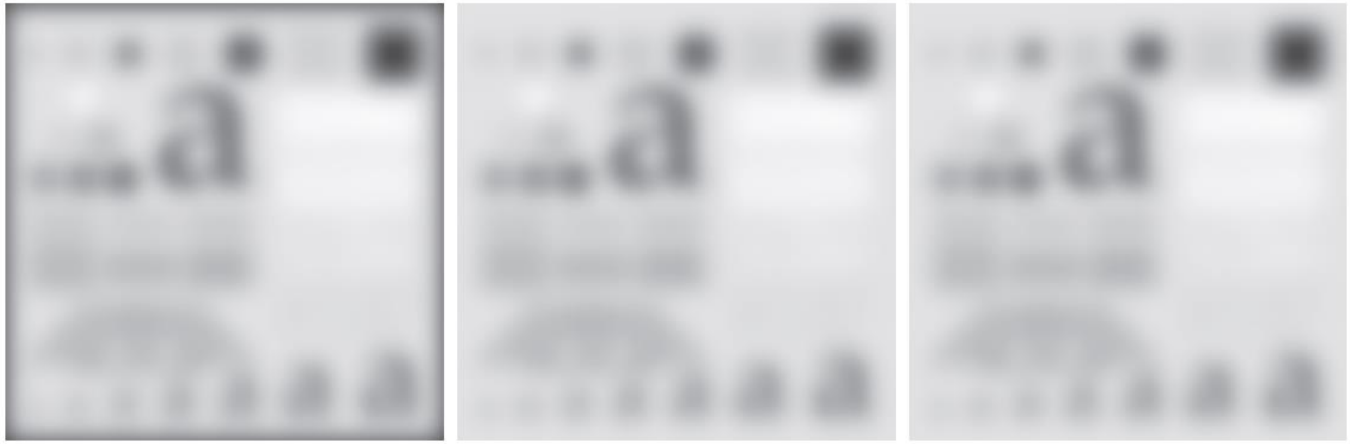
Filtering

- Padding effects

m=187 σ=31 Gauss

image 1024x1024

zero mirror replicate



- Relative size effect

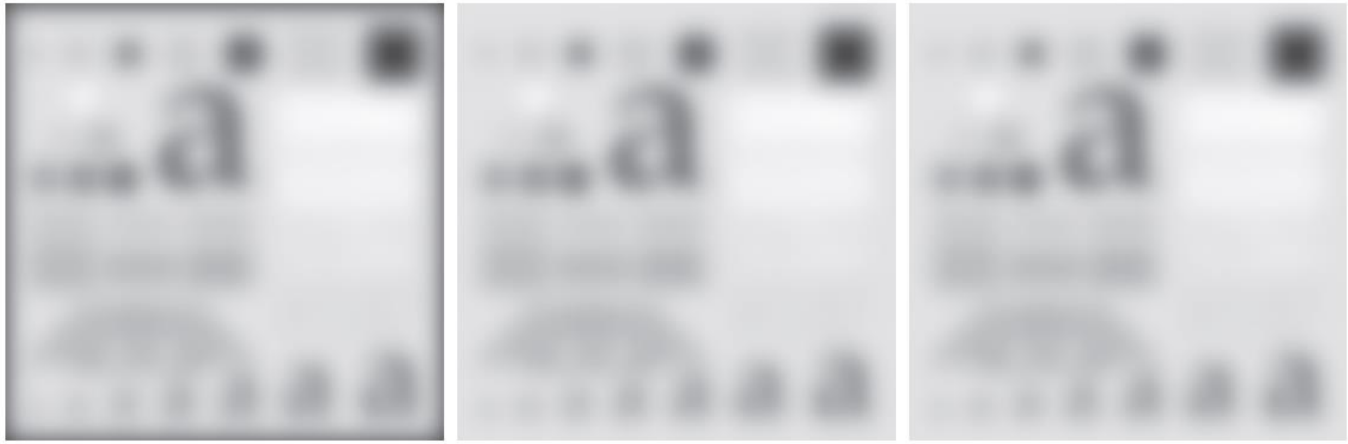
m=187 σ=31 Gauss

image 4096x4096

Filtering

- Padding effects

zero mirror replicate



m=187 σ=31 Gauss

image 1024x1024

- Relative size effect

m=187 σ=31 Gauss

image 4096x4096

m=745 σ=124 Gauss

Filtering

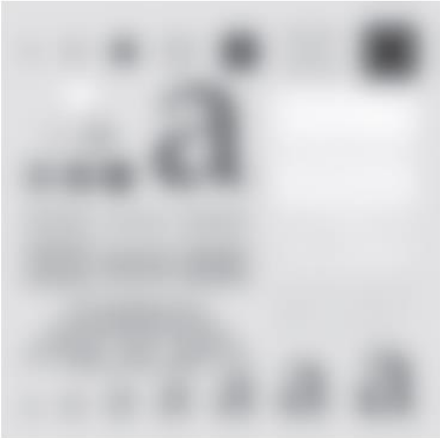
- Padding effects

m=187 σ=31 Gauss

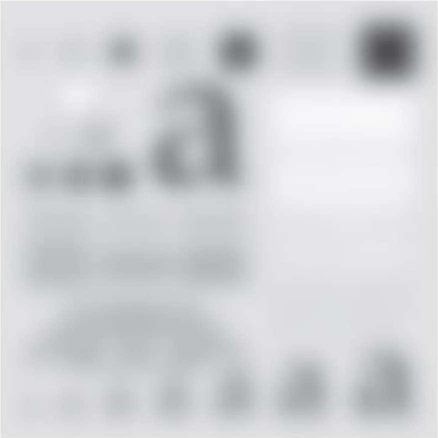
image 1024x1024

zero

mirror



replicate



- Relative size effect

m=187 σ=31 Gauss

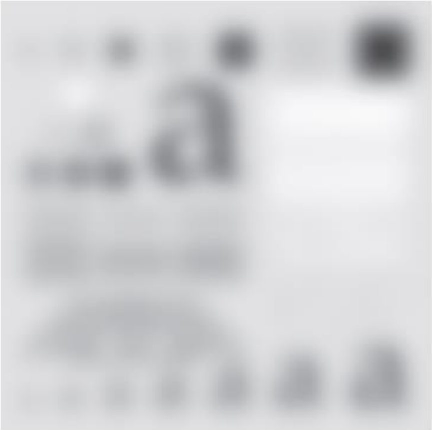
image 4096x4096

m=745 σ=124 Gauss

input

m=187 σ=31

m=745 σ=124



Filtering

- Relevant region extraction

Filtering

- Relevant region extraction

Filtering

- Relevant region extraction

Filtering

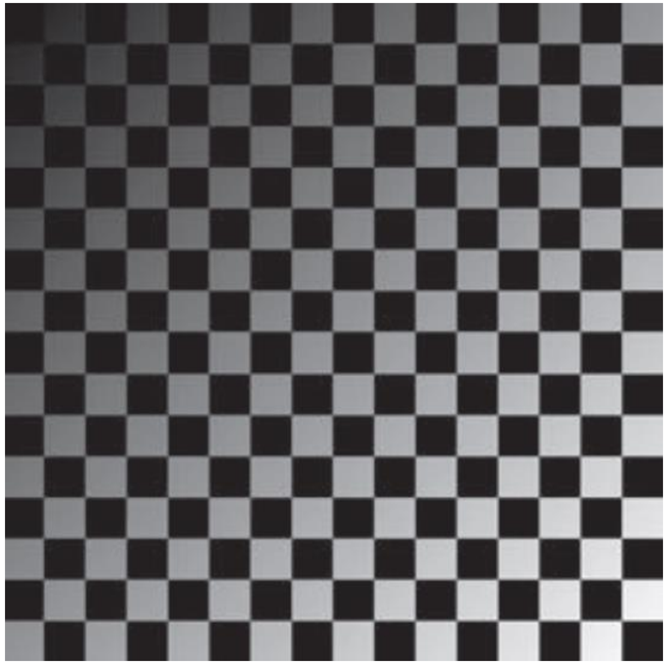
- Relevant region extraction

filtering

thresholding

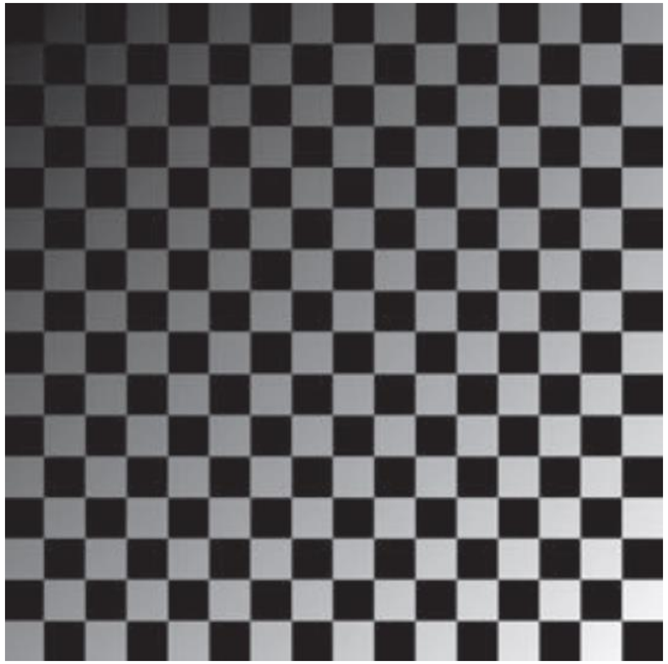
Filtering

- Shading correction



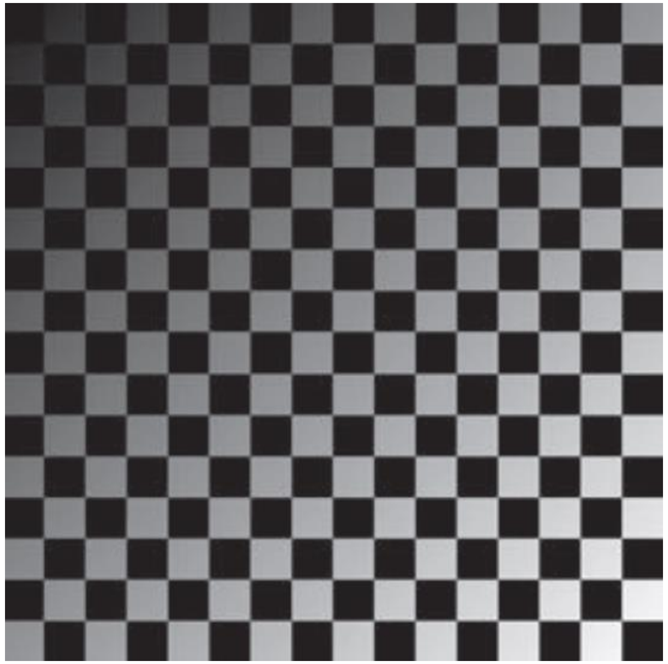
Filtering

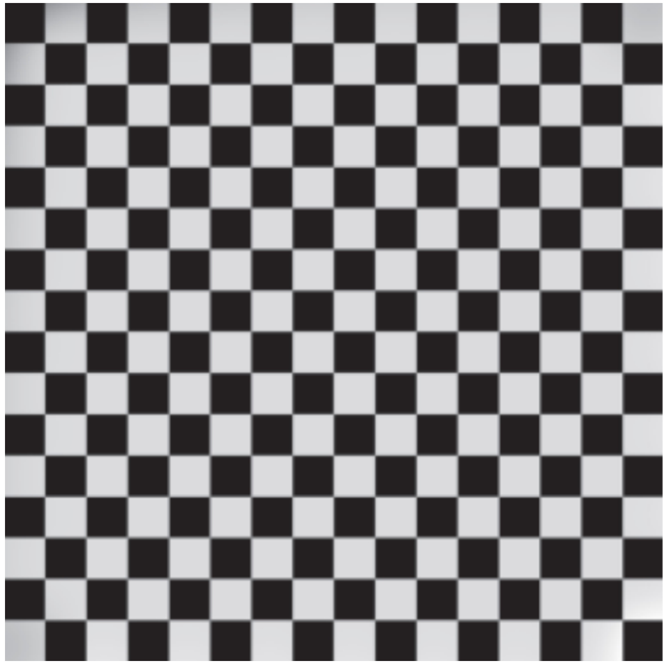
- Shading correction



Filtering

- Shading correction



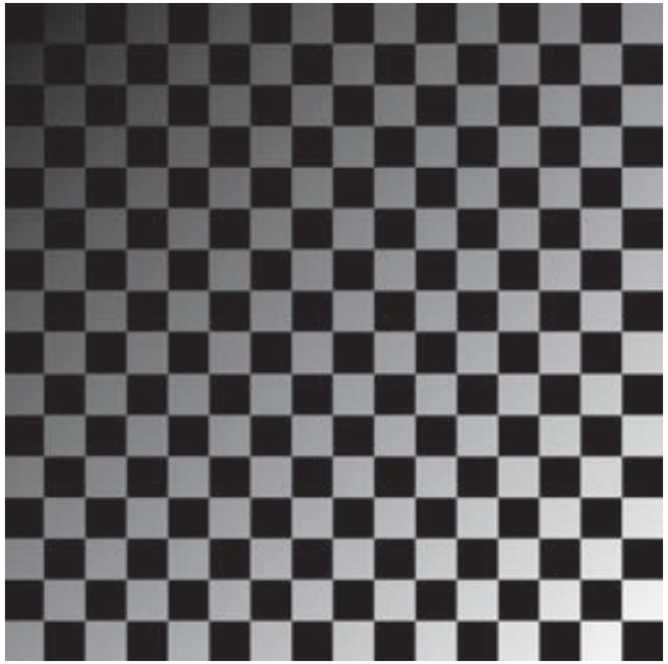


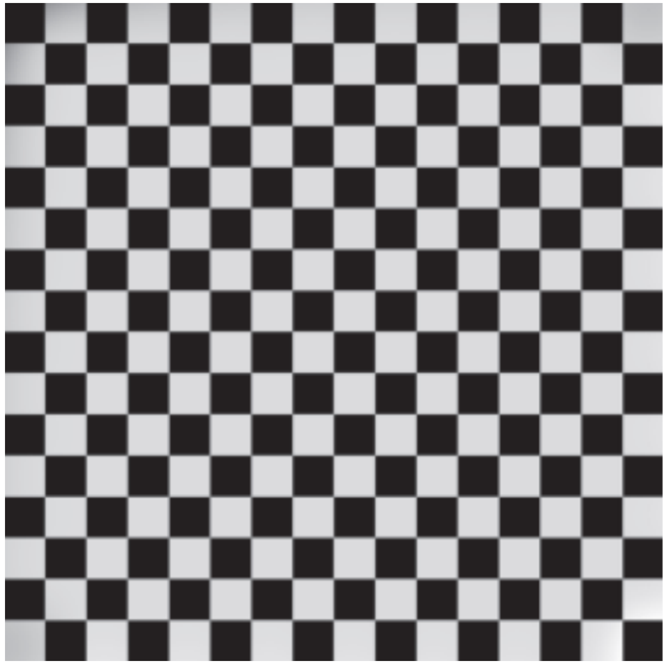
Filtering

- Shading correction

filtering

scaling





Conclusion

- Filtering
  - Separable kernels
  - Correlation Vs Convolution
  - Filter properties
  - Smoothing filters

Filtering

credit: S. Bidun'20